NASA Contractor Report 178365

ICASE REPORT NO. 87-57

ICASE

GRID GENERATION FOR THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

Peter R. Eiseman
Gordon Erlebacher

(NASA-CR-178365) GRID GENERATION FOR THE SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS Final Report (NASA) 89 p Avail: NTIS HC A05/MF A01 CSCL 09B
N87-30076
Unclas
G3/61 0099835

Contract No. NAS1-18107
August 1987

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

D

Grid Generation for the Solution of Partial Differential Equations

Peter R. Eiseman
Columbia University

Gordon Erlebacher
NASA Langley Research Center

Abstract

A general survey of grid generation is presented with a concern for understanding why grids are necessary, how they are applied, and how they are generated. After an examination of the need for such meshes, the overall applications setting is established with a categorization of the various connectivity patterns. This is split between structured grids and unstructured meshes. Altogether, the categorization establishes the foundation upon which grid generation techniques are developed. The two primary categories are algebraic techniques and partial differential equation techniques. These are each split into basic parts, and accordingly are individually examined in some detail. In the process, the interrelations between the various parts are accented. From the established background in the primary techniques, consideration is shifted to the topic of interactive grid generation and then to adaptive meshes. The setting for adaptivity is established with a suitable means to monitor severe solution behavior. Adaptive grids are considered first and are followed by adaptive triangular meshes. Then the consideration shifts to the temporal coupling between grid generators and PDE-solvers. To conclude, a reflection upon the discussion, herein, is given.

Research was supported under the National Aeronautics and Space Administration under NASA Contract No. NAS1-18107 while the first author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665. Also, research was supported under NASA Grant NAG3-715 and Air Force Grant AFOSR-86-0307.

INTRODUCTION

The numerical solutions to partial differential equations are approximations that usually depend upon some finite collection of points that cover the field on which the solution is sought. If the collection is arbitrarily given and no connections between the points are specified, then the solution depends only upon the points and is thereby free of any mesh: a mesh requires both the points and the connections between them. In the context of hydrodynamical simulations, the mesh-free objective has been pursued by a number of investigators. There, the points in the field ride along with the fluid in a Lagrangian sense. This permits long time non diffusive simulations at the expense of having to deal with a somewhat random array of points. Various approaches for dealing with such arrays are given in the conference proceedings entitled: The Free-Lagrange Method (89). The same type of construction has also spread to general problems with aerodynamic configurations (126).

In the most basic form, every point in the array of points has a unique cell about it defined by the points which are closer to it than to any other point in the array. This cellular decomposition is known as the Voronoi mesh and its dual is called the Delaunay mesh. While Voronoi cells do not provide a connectivity graph between the original array of points, the associated Delaunay mesh does. As a consequence, there is then always a mesh that is uniquely defined by an arbitrary array of points. With this definition, the Delaunay mesh might lead to some undesired intersections of triangle edges with the given boundaries of objects in the field. However, there is no real obligation to construct Delaunay meshes based on the Voronoi construction. The connectivity pattern between the points are typically adjusted either to adapt more closely to the geometry and the solution, or to accommodate a particular solution algorithm or method of mesh generation. In conjunction with finite-element methods for example, there are many mesh generators that do not employ Voronoi concepts at all. From a general perspective, the construction and use of any arbitrary array of points presents the most flexible format for adjusting to complex configurations and solutions. It also bears the burden of having the most complicated management of data. This is caused by the lack of organization: the nearest neighbors to any given point are not simply given. As a consequence, the nearest neighbors, the associated volumes, and the interfaces between such volumes must be explicitly defined by connectivity arrays that are not trivial. In addition, many of the best and most efficient numerical solution techniques are either greatly weakened or are unavailable with such unstructured meshes.

At the other extreme, there are the coordinate grids for which most of the best and efficient numerical solution techniques are available. The most highly structured grid is the Cartesian grid where the connectivity for a regular block of points is provided by line segments of fixed length that are aligned with the Cartesian directions. At a basic level, the structure is evident in terms of simply knowing the nearest neighbors to any given grid point. There is then clearly no need for the previous connectivity arrays. Aside from the structure in the connectivity pattern, the structural quality in the relative pointwise positions is also high: the spacing is uniform in each direction and the intersections between the lines are orthogonal. Altogether, with both forms of structural quality, the Cartesian grids lead to the most efficient simulations. Unfortunately, they are not flexible enough to deal with generally curved boundaries.

To bring the connectivity structure of Cartesian grids onto regions with curved boundaries, coordinate transformations are employed. The Cartesian grid is then mapped onto a physical space grid so that the boundaries of the physical region are represented by discrete coordinate curves or surfaces. This removes the need to interpolate for boundary conditions. In the mapping process, the intrinsic quality of the Cartesian structure cannot be fully preserved. Instead, it can only be approximately achieved in terms of orthogonality and smooth variations in the spacing of cell volumes. The deviations from orthogonality should then not be too great, nor should the spacing between points or the growth of cell sizes be too large. All of these structural attributes are, of course, balanced against other requirements. These include the location of boundaries and the specification of pointwise distributions thereon or angles therefrom. They also include the clustering of points, curves, or surfaces whether they are prescribed by a priori judgements or by an automatic procedure.

The balance between the desirable structural attributes and the desired constraining features is at the very heart of grid generation. It is here that the element of control must be developed so that there is some assurance that the desired objective can be obtained in terms of the stated constraints. The grid must not become so poorly structured as to render it ineffective for numerical solution algorithms. On the side of structural quality, conformal and orthogonal transformations are typically considered to be the best. In analytical terms, this can be measured by the degree of simplicity assumed by the associated metric tensor, the coefficients of which form a matrix. With orthogonality, the metric is diagonal, and moreover, with conformality, the diagonal form also has equal entries. The price for these simple metric structures is that control over the boundary geometry or pointwise distributions is relinquished.

In two-dimensions, the pointwise distributions are determined by conformal transformations and are quite restrictive in the choice of orthogonal transformations. In three-dimensions, both transformation types are severely restrictive on the choice of geometry, not to mention the pointwise distribution on the geometry. The forced situation is then the inclusion of some nonorthogonality in order to consider even some of the most basic objectives. In the competition between the stated objectives and the metric structure, the controls in the grid generator represent explicit statements for the balance between them.

The two primary categories for numerical grid generation are algebraic methods and partial differential equation methods. More specifically, algebraic methods are based on explicit algebraic formulas while partial differential equation methods involve the solution to sets of differential equations. Conformal mapping techniques lie at the root of both methods, but such methods have since evolved into more general concepts to address the more arbitrary constraints arising from practical situations. To satisfy the constraints, the above element of control over the transformations has become the most important aspect. Control can be exercised in a fixed, programmed manner, in a dynamic interactive session, and in an adaptive context. In the general applications setting, the transformations are generated with varying degrees of automation and are assembled to form grids with topologies that conform best to the given physical problems. The topological issues are typically done in a multi-block format or equivalently by a multitude of cuts, patches and opened slits. They may also be considered with overlapping grids and with separated grids being connected by unstructured meshes.

Discussions begin with the connectivity patterns for the structured curvilinear grids and for the general unstructured or partially structured meshes. Then the various grid generation techniques, the use of interactive graphics, some postprocessing procedures, and the development of adaptive grids are examined. The adaptive requirements arise when the solution experiences rapid variations on scales that are too small for fixed grids, largely because they cannot be sufficiently refined at the key locations without being refined everywhere. Under such fixed conditions, the grid contains an excessive number of points. The development of adaptivity concentrates on the use of structured grids through pointwise movement rather than on local changes in the number of points. The latter topic is addressed more thoroughly in other chapters of this volume.

The general topic of grid generation has been the subject of a number of conferences. These occurred at NASA Langley (186) in 1980, at Nashville (223) in 1982, at Houston (97) in 1983, and at Landshut, West Germany (111) in 1986. The latter is the start of a two year international conference sequence. The topic also has appeared in a conference on adaptive methods in Maryland (11) in 1983 and has assumed a role of growing importance in the AIAA Computational Fluid Dynamics Conference sequence as well as in the international series on Numerical Methods in Fluid Dynamics, published in the numbered sequence of Lecture Notes in Physics by Springer-Verlag. A special conference on conformal methods has also occurred (231).

In addition, there have been general surveys by Thompson, Warsi, and Mastin (221) in 1982, by Thompson (224) in 1984 and by Eiseman (69) in 1985. More specialized surveys have also appeared in conferences at NASA Langely and at Nashville. In the NASA Langley conference, Moretti (149) covered conformal transformations; Thompson and Mastin (219), partial differential equation techniques; and Eiseman and Smith (61), algebraic techniques. In the Nashville conference, elliptic, conformal, algebraic, and orthogonal grid generation were respectively surveyed by Thompson (222), Ives (124), Smith (187), and Eiseman (65). Arising from the First World Congress on Computational Mechanics held in Austin, Texas in 1986, there were reviews on composite grid generation by Thompson (228) and by Steger and Benek (200), on algebraic grid generation by Smith and Eriksson (189) and on adaptive grid generation by Eiseman (57). Further reviews on adaptive grid generation are available by Anderson (4) in 1983 and by Thompson (226) in 1985.

A general text on grid generation has been given by Thompson, Warsi, and Mastin (225) in 1985 and monographs emphasizing various mathematical aspects of the topic have been presented by Eiseman (62) in 1980 and by Warsi (235) in 1981. Moreover, in the present review, a substantial number of references have been given to serve as an overall guide to the activities in grid generation. These references are simply meant to represent the spirit of the subject and to indicate where some further details can be found. They certainly are not meant to be a complete and comprehensive listing.

CONNECTIVITY PATTERNS

The forms of discrete coverage for the physical field are seen to vary in the degree to which the points are organized by their interconnections. This organization is a topological structure that appears as a connectivity pattern. In the unstructured case, the mesh can be readily made to conform to complex configurations, but this occurs at the expense of a lower level of algorithmic efficiency. In an opposite sense, the structured grids lead to highly efficient algorithms but require some effort to obtain conformity with the configurations. In between these two opposite polarities, there are partially structured cases and cases where the management of structured grids are required. Our discussion will be split broadly into structured and unstructured parts and will indicate the intermediate cases at appropriate spots.

STRUCTURED CONNECTIVITY

The structured connectivity patterns come from the application of one or more coordinate transformations. Each transformation is assumed to map a Cartesian grid to the physical region, and to thereby get a curvilinear grid with the same connectivity pattern.

A Single Coordinate Transformation

When one coordinate transformation is used, complex configurations are treated with various conditions enforced on the boundaries and at selected internal locations. The boundary conditions are applied to fix the points on a body, to move points along a body, to form branch cuts and patches, or to move boundary points according to some general rule. On the bodies, the points are usually fixed unless there is a desire for a reduction in grid distortion or for a stronger conformity with rapid solution variations. Off of the bodies, artificial boundaries are created in the form of cuts or patches for the treatment of regions with topological complexity. Within the physical field, these appear as internal transmissive coordinate curves or surfaces, or segments of curves or surfaces. When the physical boundary is either changing shape or is in rigid motion, the points are being driven by some general rule. Some examples, here, are the motion of water waves (217,216,114,113,7,247) and the flight of insects (112). In terms of differential equations, the above requirements are respectively translated into Dirichlet, Neumann, re-entrant (periodic or continuity) and mixed boundary conditions which are applied to the physical space position variables along the boundaries of the Cartesian domain.

The internal conditions contain the same conditions that are applied at boundaries and add to them the action of opening slits. A slit is formed by continuously pulling apart a coordinate curve or surface segment to form a local area or volume that is not covered by the grid. The uncovered region either is considered as yet another solid body in the region, or is covered with another grid. In the latter case, the grid within the opened slit may be optimally aligned with an internal body. A further advantage with local coordinates for the internal body is that the coordinate singularity caused by the slit is pushed away from the body. To construct a slit, the coordinate curve or surface segment must be given double values so that it can be separated into a top and a bottom which are then moved apart from initial overlying segments. In terms of the premapped Cartesian domain, two distinct values of physical position appear over the same Cartesian segment. The same use of double values also applies to any solution of partial differential equations over that grid. The reasons are the same.

With one coordinate transformation, the various cuts or patches are readily employed to make the grid conform to many bodies as if they were just one large body (215). When applied together with slits, a great variety of configurations can be analyzed. In general, the internal conditions can be used to carve out any number of pointwise blocks to thereby create holes in the grid and to also change the effective boundary. The creation of holes is basically an extension of the process of slit opening: the main distinction is that the holes come from areas or volumes in the premapped Cartesian grid rather than from segments of coordinate curves or surfaces. Thus, the double values for slits are replaced by single values around the area or volume. The most general grid from a single transformation is then obtained from mapping a Cartesian block of grid points where the applied conditions effectively remove a number of subblocks, provide cuts or patches, and can open any number of slits.

Multiple Coordinate Transformations

While considerable generality can be achieved by imposing various conditions on a single transformation, a more flexible format for the treatment of arbitrarily complex configurations is the use of more than one coordinate transformation. By this we do not mean the application of a sequence of transformations for the whole region. Instead, we consider separate transformations for the various parts of the region. That is, a number of distinct Cartesian grids are mapped into the physical region and are assembled to fully cover it with a mosaic of distinct curvilinear coordinate grids. The assembly can occur with varying levels of continuity.

Completely discontinuous assembly. The lowest level is the completely discontinuous case where the grids can overlap each other as well as various bodies in the field. The general application with overlapping grids has been called a Chimera scheme and consists of forming separate grids for each body in the field. The strongest development has occurred with Steger, Dougherty, Benek and Buning (205,18,19) and could be viewed as an extension of the earlier work by Atta and Vadyak (8,9), Starius (196,197,198), Wedan and South (239), and Kreiss (133). Still earlier, the basic overlap idea was also discussed by Eiseman (58) but was not tested numerically. The fundamental management of overlaid grids depends upon a means to neutralize points that fall within solid bodies and to transfer data between the grids so that boundary conditions can be applied. In the work of Steger and his colleagues, this is accomplished respectively by blanking out points in solid bodies and by local multilinear interpolation. Their applications have been in two and three dimensions and have included motion.

While the simple local interpolation is successful for most of the physical field and most of the practical simulations, improvements can still be made so that accuracy is increased. This comes from a consideration of characteristic directions and conservative properties. The characteristic boundary conditions for overset grids were studied by Eberhardt and Baganoff (55) and had led to an improved passage of shock waves through the grid overlap. More improvement can be expected from a correct accounting of conserved quantities and this was pursued by Berger (22,20).

With the view of adapting the overlapping grids to the solution rather than to the geometric components of some complex configuration, a parallel development has been undertaken. The basic study was presented by Berger and Oliger (23) who considered solution refinement by a succession of solution oriented Cartesian grids. This was also employed by Berger and Jameson (21) with the orientation restricted to Cartesian directions that are mapped into coordinate directions about an airfoil. Further work in the general context has also been pursued by Oliger (159), Venkatapathy and Lombard (232), Lombard and Venkapathy (136), Henshaw and Chesshire (115), Reyna (172), Henshaw (116), Chesshire (37), Brown (32), Fuchs (92), Mastin (141), Thompson (220), Mastin and McConnaughey (140) and Rai (169).

In the absence of rigid geometric constraints, the individual grids of an overlapping collection are rather simply generated. The most [illegible]manding requirement here is just to generate grids about simple bodies or simple components of more complex bodies. Away from the bodies or components, the corresponding grid is relatively unconstrained and thereby leads to the simplicity in its generation. This, however, amounts to shifting the complexity from the grid generation to the boundary conditions. As a consequence, we get boundary condition strategies that involve interpolation, characteristic directions, and conservation principles. In addition, we must also worry about the relative cell sizes. Significant error can arise if the data transfer occurs between very fine and very coarse grids. In a sense, there is then a weak implicit constraint that cell sizes at the overlap locations cannot be too disparate. From a rough perspective, we may now suitably summarize the use of overlapping grids as being the most flexible format for grid generation and the least flexible format for boundary conditions.

Partially discontinuous assembly. In leaving the overlapping grids we go from the completely discontinuous assembly of grids to a partially discontinuous assembly. The main constraint on the grids is that distinct grids are joined together along an interface but are not required to match in a pointwise sense on the interface. In three-dimensions, an intermediate level of discontinuity can also occur when one of the two families of coordinate curves on the interface is aligned but the other is not. As can be expected, the interface constraint causes some shift in difficulty from the boundary conditions to the grid generation. The conservation of physical quantities is readily enforced on cells that are centered at the interface. In two dimensions, these cells are constructed by forming half cells with respect to the grid on one side and then obtaining full cells by extrapolation to the median between the interface and the first coordinate curve on the other grid. By carefully balancing fluxes across the edges of such cells, the quantities inside are conserved. Particular attention here is required at the median curve.

The method of conservatively treating the interface is due to Rai (169,170,171,168) and is quite effective in fluid mechanics problems. The studies have been in two-dimensions and should be extensible into three dimensions. Severe disturbances have successfully been passed through the interfaces even when one of the grids moves relative to the other along the interface. The use of motion in this context may be the best means of treatment when the simulation is for a device that has moving parts. Clear examples are helicopters, propeller driven aircraft and multistage turbine blade configurations.

Continuous assembly. When grid point alignment is required along interfaces, the assembly of grids is accomplished in a continuous fashion but may have derivative discontinuities at the interface. At this stage, the boundary conditions are considerably simpler to apply while the grid generation is only slightly more difficult. Although no interpolation is required at the interface, the transition across it must still be done carefully to account for discontinuous derivatives. Meanwhile, the main added difficulty for grid generation is the requirement of alignment. This may cause excessive skewness if it is done poorly. However, it is usually the case that a relatively good alignment is easy to achieve because great precision is usually not required for this task. Some examples of this are given by Eiseman (59), Forsey, Edwards, and Carr (83), Kowalski (131), Ghia, Ghia and Studerus (96), Steger (201), and DiCarlo, Piva and Guj (47).

Derivative continuous assembly. Further enhancement in the continuity of assembly comes with the enforcement of derivative continuity in increasing levels. In the discrete sense of grids, first and second derivative continuity are the only levels of practical interest. In many cases, only first derivatives are needed since a reasonably well-defined three-point stencil is produced through the interface. Derivative continuity beyond this level leads to a more modest change in the grids. The last trace of fairly visible continuity occurs with second derivatives in three dimensions where the observed variation is the deviation of a curve from being planar. As a consequence, derivative continuity that exceeds the first order can be assumed to be reasonably complete for grids. Moreover, it does not make any sense to consider levels of interface continuity greater than that of the grids being assembled. However, it is often sensible to consider intermediate cases with partial continuity in a derivative. For example, angles may be matched at interfaces while the grid spacing is not. The reason for this is that an improved cell structure can still be readily seen in the grid at the interface.

With the various levels of continuity beyond the completely discontinuous overlapping case, an interface is always present between the distinct systems being assembled. In each instance, there are two options for establishing the interface: it may either be specified or be determined by the grid generator. When it is specified, it is usually called a patch boundary. This term is used to indicate that a lower underlying level of derivative continuity is present due to a required explicit construction. As has been noted, the noticeable effects are ultimately taken from the grid rather than the analytical basis and are thus fairly decent with first or second derivative continuity. In most cases, only first derivative continuity has been employed. The first such developments were given by Eiseman (60), Smith and Weigel (188), and Sorenson and Steger (191). By contrast, when the interface is determined by the grid generator, it is treated like any internal coordinate curve or surface within the coordinate systems. As a consequence, it inherits the same continuity level as the given coordinate transformations. With the view of extending the terminology of conformal mappings, this interface is called a branch cut. In the more general context, grids that are generated from elliptic partial differential equations often approximate an analytical solution which is infinitely differentiable as originally discussed by Thompson, Thames and Mastin (215). The branch cut then typically comes from the use of a three point difference stencil in each coordinate direction. Although, an underlying infinite level of differentiability is often present, it is not of great importance for the grid since typically the derivatives are all computed numerically. The assurance, however, is that complete continuity in the sense of grids is achieved quite simply. Beyond this, the main advantage is that a detailed and perhaps tedious interface construction is not needed.

Block structured grids. When some level of continuity is present across interfaces, the overall assembly of distinct grids is called a composite or block structured grid. The general pattern has been simultaneously developed by so many investigators that it is difficult to identify any one of them as being the first. This may be quite natural with such an obvious pattern and can certainly be detected in many of the works presented in the grid generation conference of 1980 (186), not to mention the previous works. In the evolution of the block structured approach, a basic classification of the developments has arisen and comes directly from the boundary conditions for the block edges or faces. Quite simply, the boundary conditions on each edge or face are either restricted to be of only one type or they are not. Under the restriction, the physical regions must be dissected into more coordinate zones than would have otherwise occurred had various distinct conditions been permitted. For example in two-dimensions, when a branch cut leaves a solid body, the restriction would imply that the body and the cut must be treated by separate adjoining blocks. The corresponding conditions are the specified points on the body and the continuous transmissive interface for the cut. If the coordinates are to wrap around both the body and the cut, then the grid is called a C-type grid and three separate blocks are required: one above the cut, one around the body and one under the cut. By contrast, had the body and both sides of the cut been permitted on a single block edge, then only one block would have been required. Beyond this modest case, the disparity in the number of required blocks is substantial in two dimensions and is even more dramatic in three dimensions.

While the increased number of blocks may appear to be cumbersome, the restriction which caused the increase represents an organizational advantage. With a single boundary condition type on each edge or face, the connectivity pattern of the entire configuration is more simply stated. The corresponding integer arrays involve the block number and size, the connectivity between blocks (oriented faces of adjoining blocks), and the boundary condition index. Such arrays have been called topology files. Some prominent examples of this topological organization are given by Coleman and Brabanski (39), Rubbert and Lee (175), Boerstoel (27), Shaw, Forsey, Weatherill and Rose (185), Sawada and Takanashi (180), Seibert (182), Roberts (174), Fritz (90) and Fritz and Leicher (91).

With the alternative category, the main advantage is a reduced number of blocks while the disadvantage is the expense of a more complicated arrangement of boundary conditions. The complications arise from the need to segment block edges or faces and then to label each such segment with indices for the connectivity and the boundary condition type. To provide full generality, there must be enough room for many segments on each edge or face. the number of segments must then be determined and is seen to multiplicatively expand the number of required indices. The total number of indices for the entire block structure, however, will be less than that required of the previous category. This occurs since boundary segment indices would be repeated when larger blocks are subdivided into the smaller ones that conform with single boundary condition types on edges or faces. The inconvenient feature here is that the topology file must deal with the possibility of arbitrarily many segments. Prominent examples of this topological organization are given by Thompson (227), Berglind (24), Soni (190), Eiseman (66), Ghia, Ghia and Ramamurti (98), Eriksson (76) and Sorenson (193,194).

In applications, it is convenient to form the blocks with an extra layer of points beyond the actual boundaries. Although more computer storage is consumed, each block contains its own current boundary condition data and can, thereby, be effectively separated from the others. The separation is particularly helpful in the organization of the simulation and/or the grid generations and provides a decent data structure for parallel computing environments. A more detailed account of extra layers can be found in Thompson, Warsi and Mastin (225), in Coleman (38), and in Miki and Takagi (144,145).

Because of the additional storage and the consequent double fetching from the storage, Coleman and Brabanski (39) have replaced the extra layers with an equivalent indexing strategy to define difference stencils on block boundaries. The consideration arose in the three-dimensional context where the added cost appeared to be too much. As a result, they cut the cost at the expense of having a more complex algorithm. Although the demands of an extra layer may appear to be too much in three-dimensions, it should also be noted that such demands should be compared to the overall sizes. When the blocks are small, the extra layer represents a large addition while the opposite is true for large blocks. When N points appear on each edge of a three-dimensional block, there are $(N+2)^3 - N^3$ points in the extra layer surrounding the basic block of N^3 points. A meaningful measure of the additional storage is given by the fractional increase in the number of points caused by the extra layer. This is just $(1+2/N)^3 - 1$. When the fraction is given by $1/m$, the necessary integer value of N is just $2 + 6m$. Thus, the extra layer represents an equal amount of storage when $N = 8$, is half the storage when $N = 14$, and so on. As a consequence, when the minimum number of points on a block edge is 8, the extra layer has at most the same number of points as in the original block and we have at worst not quite doubled the storage. Similarly, when the minimum number is 14, we have at worst a 50% increase in storage; and so forth. By the time we reach 50 points, the increase is at most only 12.5%. The main conclusion that can be drawn from these numbers is that the extra layers are more likely to be useful when distinct boundary conditions are permitted on faces since then, as we have noted, the blocks are larger.

Singularities. As a final note, the general consideration of grid topologies gives rise to an assortment of coordinate singularities regardless of whether or not block structures are employed. A discussion for the various types of singularity together with strategies for their treatment are given by Eriksson (78), Mastin and McConnaughey (140), and Thompson, Warsi and Mastin (225).

MORE GENERAL MESHES

As computers evolve into ever more powerful machines, with increased storage and computational power, the problems people wish to solve become increasingly complex. This complexity is present both in the physical content of the governing equations and in the more realistic physical configurations that are of interest. When the geometrical complexity of the problem dramatically increases, the task of preserving a good mesh becomes correspondingly more difficult. The constraints imposed by near orthogonality of the grid both in the interior and on the boundary of the physical domain, greatly lengthen the amount of time required to produce the initial grid. Furthermore, as the solution evolves, it is quite likely that the salient features of the flow do not remain fixed in space. Therefore, mesh topologies that allow easy restructuring of the mesh become highly desirable. Although the use of a general connectivity mesh poses more difficulty, this is likely to be somewhat offset by advances in computer architecture and numerical solution strategies.

There are a multitude of mesh topologies which are suitable for the space and time evolution of complicated configurations. These are now discussed in more detail.

Triangular Meshes

Any two-dimensional physical domain, including those with multiple bodies can be discretized with a finite collection of points. These points uniquely determine a set of non-overlapping cells, which are collectively known as a Voronoi mesh. More precisely, a Voronoi cell about a node c is the set of points closer to c than to any other node. The bissectors of the edges of the Voronoi cells form a triangular grid which is the dual of the Voronoi mesh. It is called a Delaunay triangulation and has some very interesting properties. On such a mesh, internode connections generate triangles which are the nearest to equilateral in some average sense as witnessed by Dukowicz (49). This result is based on an important property of Delaunay meshes, namely that the circumcircle of each triangle does not contain any other mesh point. This eliminates the likelihood of long, thin triangles since the circumcircle radius then becomes very large. Algorithms based upon Voronoi meshes have been developed by Augenbaum (10) and Börgers and Peskin (28). Voronoi meshes are also defined in three dimensional spaces, and several studies have taken advantage of this. These include the works of Trease (229,230) and Jameson, Baker, and Weatherill (126).

When the physical domain includes bodies with scale lengths smaller than the local mesh point interspacing, it is possible for the resulting Delaunay mesh to produce undesirable side effects. For instance, mesh edges might intersect the wing of an airplane. One must therefore locally restructure the grid in these areas, either manually or with some form of automated procedure.

Several automated triangular mesh generators have been developed for general domains. An excellent list of references covering the period up to 1980 is provided in the review by Thacker (211). There is no restriction on the complexity of the domains since the triangle edges can conform to arbitrary curved boundaries. Furthermore, problems with dynamically changing topologies become easier to handle because the grid can always be locally restructured to insure that the triangle edges are parallel to the newly formed boundaries. Boundary conditions must only be derived on a general boundary cell. They are then applied to all boundary cells. For example, Neuman boundary conditions are expressed by a specialized formula relating the affected node to its nearest neighbors. These formulas allow an arbitrary number of edges to intersect at a node. Therefore, they can indiscriminately be applied to every boundary point where Neuman conditions are valid. Such formulas are reduced cases of the general expressions for differential operators considered in detail by Erlebacher (81).

The advantages cited above are somewhat tempered by several weaknesses. Because mesh points are distributed and numbered in a random fashion, it is necessary to keep track of a number of geometric quantities. Edge, triangle and node information must be stored and kept current as the mesh evolves. For example, in a finite-volume code, the triangle numbers adjacent to each edge must be known. Similarly, in a node-centered scheme, it is the vertex numbers of each triangle which become important. Once the appropriate data structure is constructed and translated into a set library of computer modules, it becomes relatively easy to manipulate the grid elements. Similar data structure considerations are emerging in problems which call for block-structured grids.

Another potential disadvantage is the difficulty in constructing high-order numerical finite-difference or finite-volume schemes on triangular grids. This difficulty is somewhat diminished by formal finite-element formulations which can, in a systematic fashion, generate schemes of arbitrary order. General discussions are available in Strang and Fix (210) and in Oden and Reddy (157).

There is no direct correspondence between a totally unstructured grid and a uniform distribution of points. Consequently, implicit schemes which typically require the inversion of large systems of linear equations, is not directly feasible. Nonetheless, multigrid techniques which do not depend on a nested structured grid have been developed by Lõhner and Morgan (135) and Mavriplis and Jameson (142).

Vector algorithms specialized to unstructed meshes , long thought to be very difficult have made their appearance. The recent hardwired gather and scatter routines available on most supercomputers have made this possible. These routines allow a selected set of randomly distributed data to be organized into a linear vector array which can then be processed in the usual manner.

Unstructured triangular grids are particularly well suited to local addition and subtraction of nodes. New triangles are created by several methods. A new node inserted at the triangle centroid and joined to the three triangle vertices creates two more triangles and three more edges. Three additional nodes placed at the edge midpoints of a triangle and joined to each other create three more triangles and three more edges. However the new nodes must now be joined to adjacent triangles to maintain the overall triangular structure. Deletion of nodes is accomplished by the reverse process.

As a triangular grid dynamically evolves in time, distorted triangles make their appearance. These are eliminated by a process called mesh restructuring. For example edge flipping consists of treating two adjacent triangles as a quadrilateral. The offending diagonal is removed and replaced by the other when certain conditions are met. For example, the sum of the angles subtended by the new diagonal should be greater than the total angle subtended by the old one. Another criteria might be that the shorter of the two diagonals is always chosen.

Structured triangular meshes have been considered for their similarity to their curvilinear counterparts. It is possible to construct a triangular grid which can smoothly map onto a grid of equilateral triangles. This approach has successfully been implemented by Winslow (245).

It is possible to construct triangular meshes which possess a moderate level of structure. Unfortunately, some of the flexibility achieved by unstructured meshes is lost. One of the earliest structured grids was used by Winslow (245) in his study of magnetic fields in a multi-component system with complex boundaries. Each node lies at the intersection of three coordinate lines. Therefore there is a one to one correspondence between his mesh and a uniform mesh of nearly equilateral triangles. This provided a good setting for the use of a Laplace solver to generate the grid. That generator set the stage for the development of the more general elliptic methods of grid generation. Local grid refinement and restructuring were of course forbidden for otherwise the structure would be lost.

An alternate approach is to cover the physical domain with a coarse unstructured mesh. Coarse triangles are referred to as macro elements or macro triangles. Each macro element can then be subdivided into four smaller triangles, each congruent to the original element. This is done by joining the edge midpoints to each other. A second application of this process to the four newly created elements generates 16 new triangles. This nesting procedure can be individually applied to every triangle independently. Interpolation algorithms become necessary to transfer information across macro element boundaries. Such a scheme reduces the computer storage requirements because the structure in each macro triangle is regular. Local refinement (or coarsening) is limited by the macro triangle level, and restructuring is forbidden. If restructuring were permitted, one would be forced with transferring information between overlapping sets of triangles. When the problem under consideration is time-dependent, this could lead to unacceptable levels of numerical dissipation. Multiple levels of nested regular triangular grids are well-suited to multigrid schemes where the successive grid coarsenings consist of removing successive layers of the previously refined mesh. This partial structure has been suggested by Eiseman (70) for a global moving mesh and by Bank (15) for the local refinement of a static mesh. An alternative procedure was developed by Marshall, Eiseman and Kuo (137) wherein a full rectilinearly ordered grid is inserted in each macro element in such a way as to provide a uniform distribution of points within such macro elements. This was done for both two and three dimensions.

Mixed Elements

Nakahashi (153) considered multiple body configurations. Well-structured (i.e. nearly orthogonal) curvilinear grids are generated around each body. But rather than try to match the individual grids at some common interface, buffer regions are created and are triangulated. Finite volume differencing is used on the curvilinear grids, while finite-element methods are applied to the triangulated areas. Convergence of the numerical scheme is achieved by iterating back and forth between solutions on the curvilinear grids and the triangulated buffer regions. Information is transferred through the region interface nodes. In a sense, this approach contains some of the aspects of the Chimera schemes (205,18,19). It provides a method to transfer data between structured grids through unstructured buffer regions instead of through a direct transfer on overlapping regions. If the physical bodies are close to one another, the construction of the buffer zones is not always clear cut. This approach may become impractical in situations where the bodies are moving relative to each other (as in the store separation problem). In comparison, the Chimera scheme is simpler to implement albeit the transfer of information across the interfaces is more complex.

On the other hand, Crowley (43) has developed a Free-Lagrange, two-dimensional code in which the vertices are convected with the fluid. Refinement and restructuring occur locally to preserve a non-distorted grid. Simulation generally begins with a curvilinear grid, which progressively acquires increased triangular cells as local regions are restructured, coarsened or refined. The inclusion of triangular elements avoids the need for global rezoning. Special care must be taken to avoid concave quadrilaterals during the reconnection process. Otherwise, an overlapping grid becomes highly probable. Crowley (43) and Cooper (41) provide many details. A similar acquisition of triangular cells occurs with front tracking. The triangles are localized to the front region, leaving the remainder as well structured quadrilateral cells. A description is given by Glimm et al. (100) and Chern et al. (36).

General Connectivity In Three Dimensions

Three-dimensional general connectivity patterns have been considered by Fritts (88), Marshall, Eiseman and Kuo (137), Trease (229,230) and Jameson, Baker and Weatherill (126). In the context of Free-Lagrange methods, Fritts (88) has generalized two-dimensional restructuring algorithms to handle arbitrary tetrahedral configurations.

GRID GENERATION TECHNIQUES

The two primary categories for numerical grid generation broadly correspond to explicit and implicit definitions in the sense that one is directly specified by equations while the other comes only from the solution of equations. The equations in the explicit case are just algebraic formulas and in correspondence, the category is called algebraic grid generation. The equations in the implicit case are partial differential equations and accordingly the methods are called PDE-methods. These are further split into the subcategories entitled elliptic, hyperbolic, and parabolic grid generation to reflect the type of PDE used to generate the grids.

Having covered topological issues in our discussion of connectivity patterns, we shall restrict our attention here to a single transformation and concentrate upon the various aspects of control. The issue of control arises in the general applications setting when the desired problem parameters are set forth and result in a number of constraints to be placed upon each coordinate transformation. The capability to successfully satisfy such constraints is the ability to control the grid. Attention will be given to the construction of the controls and to how, when, and where they are applied.

METRIC NOTATION

In the discussion of the various grid generation methods as well as in the various adaptive strategies of the next section, the fundamental properties of each object can be more clearly and concisely stated with the use of metric notation than without it. As a consequence, we shall describe just enough of it for our purposes. In two dimensions, we associate 1 with ξ and 2 with η. The differential element of arc length ds is then given by

$$ds^2 = g_{11}d\xi^2 + 2g_{12}d\xi d\eta + g_{22}d\eta^2 \tag{1}$$

where

$$\begin{aligned} g_{11} &= \vec{r}_\xi . \vec{r}_\xi &&= x_\xi^2 + y_\xi^2 \\ g_{12} &= \vec{r}_\xi . \vec{r}_\eta &&= x_\xi x_\eta + y_\xi y_\eta \\ g_{22} &= \vec{r}_\eta . \vec{r}_\eta &&= x_\eta^2 + y_\eta^2 \end{aligned} \tag{2}$$

and $\vec{r} = (x, y)^T$ is the position vector. This form of ds^2 can be readily verified by a chain rule expansion of $dx^2 + dy^2$. The matrix of elements g_{ij} is called the metric because it defines the distance measurements with respect to the coordinates (ξ, η). In a straightforward manner, the determinant

$$g = det(g_{ij}) \tag{3}$$

is also seen to be the square of the Jacobian

$$J = x_\xi y_\eta - x_\eta y_\xi . \tag{4}$$

For grids, this gives the cell sizes in the physical (x,y)-space that correspond to the fixed uniformly sized cells in the computational or logical space (ξ, η). When the Jacobian is nonvanishing, the inverse of the metric matrix (g_{ij}) can be computed and has elements that are typically written in the superscripted format g^{ij}. In continuation, the metric notation is equally valid in higher dimensions. For three-dimensions, we merely associate 3 with a third variable ς and then repeat the above discussion where, of course, the expressions are slightly longer but the meaning remains the same. Further discussion can be found in virtually any text on differential geometry. A development specifically aimed at grid generation can be found in either of the monographs by Eiseman (62) and Warsi (235).

ALGEBRAIC TECHNIQUES

Algebraic grid generation distinguishes itself from other grid generation methodologies by the ability to provide a direct functional description of the coordinate transformations between the computational and physical domains. The roots of algebraic grid generation are found in conformal mapping, defined by explicit analytic functions of a complex variable. Conformal maps have found many early applications in potential flow. Although the resulting grids preserved a basic cell shape, the amount of local control provided was not sufficient for many problems. One by one, the intrinsic properties of conformal mapping were dropped. Cell shape preservation was replaced by only orthogonality; thus allowing the grid to stretch in one or more of the coordinate curve directions. Extensions to three-dimensions were developed which combined two-dimensional conformal mappings with algebraic stretchings in the third dimension; thus, producing some nonorthogonality.

Shearing transformations also remove some of the shortcomings of conformal mappings at the expense of orthogonality. The amount of control is increased but is still insufficient.

These mappings allow the specification of boundary node locations and depend only on such locations. Therefore the orthogonality of the coordinate curves normal to the boundaries cannot be assured. Hermite interpolation on the other hand is a function of the boundary node positions and the desired derivatives from the boundaries. In a classical sense, these mappings are global polynomial mappings between two opposing boundaries. More precise control of the coordinate curves in the interior regions is provided by the multisurface transformation which contains the aforementioned transformations as special cases.

When two or more pairs of boundaries must lie on coordinate curves (or surfaces), one or more different unidirectional coordinate transformations can be combined using Boolean sums. The resulting mappings are often called transfinite.

Although not usually considered an integral part of algebraic methods, certain post-processing techniques can also be expressed in terms of explicit functional dependencies. Movement of coordinate lines according to specific clustering requirements and the blending of independently generated grids are two examples.

In a logical progression, the discussion starts with conformal mappings, and is followed by shearing and Hermite transformations. The establishment of arbitrary shape control over the coordinate curves that connect two opposing boundaries is considered next and is obtained with the more general unidirectional construction known as the multisurface transformation. Unidirectional transformations are then combined with Boolean sums to generate the most general multi-dimensional transformations whose coordinate lines in physical space conform to all boundaries. The section ends with a few examples of grid postprocessing.

Conformal Mappings

Conformal mappings are among the first algebraic transformations used for grid generation. To simplify matters they are often applied along with direct shearing. In its most simple form, a conformal mapping is simply an analytic function of a complex variable. This variable is denoted by $z = x + iy$, where $i = \sqrt{-1}$ and the real and imaginary parts of z are the two-dimensional Cartesian coordinates (x, y) in physical space. The general mapping

$$z = g(Z) \tag{5}$$

relates the points (x, y) in physical space to the points (ξ, η) in computational space, where similarly $Z = \xi + i\eta$. In a real-valued differential form, Eq. 5 becomes

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta \\ -x_\eta & x_\xi \end{pmatrix} \begin{pmatrix} d\xi \\ d\eta \end{pmatrix} \tag{6}$$

where use has been made of the Cauchy-Riemann conditions

$$\begin{aligned} x_\xi &= y_\eta \\ y_\xi &= -x_\eta . \end{aligned} \tag{7}$$

Further relations can be derived from Eqs. 7. The metric coefficient g_{12} is zero, and the metric elements g_{11} and g_{22} are equal and denoted by h^2. To bring out the shape preserving properties of conformal mappings, it is instructive to rearrange the previous system of equations. To this effect, consider the angle θ defined by

$$\begin{aligned} \cos\theta &= \frac{dx}{\sqrt{dx^2+dy^2}} &= \frac{x_\xi}{h} \\ \sin\theta &= \frac{dy}{\sqrt{dx^2+dy^2}} &= \frac{x_\eta}{h}. \end{aligned} \tag{8}$$

With the Cauchy-Riemann conditions, the standard rules for the trigonometric functions are readily seen to satisfy the conformal metric relationships

$$x_\xi x_\eta + y_\xi y_\eta = 0 \tag{9}$$

and

$$h^2 = x_\xi^2 + x_\eta^2 = y_\xi^2 + y_\eta^2 \tag{10}$$

After the relations defined by Eqs. 7 and 8 are substituted in Eqs. 6, the differential coordinate transformation can be written as

$$\begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} h & 0 \\ 0 & h \end{pmatrix} \begin{pmatrix} d\xi \\ d\eta \end{pmatrix}. \tag{11}$$

Equation 11 can be locally interpreted as a dilation and a rotation. The dilation comes from the scaling factor h while the rotation is given by the angle θ between the curve in ξ and the x-axis of the z-plane. The magnitude of the scaling gives the local cell sizes. In an explicit sense, the angle preservation between distinct intersecting coordinate curves is displayed. Moreover, if $d\xi$ and $d\eta$ are equal in magnitude, so are the length elements dx and dy. Thus a lattice composed of square cells in computational space, transforms, under an analytical complex mapping, into a grid of quasi-squares in the physical domain.

Some examples of simple analytic functions applied to grid generation are given by the parabolic and the Joukowski transformations which are

$$z = Z^2 \tag{12}$$

and

$$z = Z + \frac{1}{Z} \tag{13}$$

respectively. The parabolic map is typically employed to unwrap the positive x-axis by sending it onto the entire ξ-axis. On the other hand, the Joukowski transformation maps certain circles in the Z complex plane to Joukowski airfoils in the z complex plane (162). It also maps the unit circle about the origin to the slit from -2 to 2 along x.

Conformal mappings are not limited to planar surfaces. Stereographic and Mercator maps are conformal mappings from the sphere to the plane. They preserve local angles, and the local scale factor is independent of orientation. A three-dimensional impeller was gridded by Senoo (184) using conformal mappings.

When confronted with a complicated geometry, one wishes to transform to a more simple configuration where the problem can be more simply solved. The appropriate conformal map must then be found. A comprehensive list of mappings was compiled by Kober (130). If the desired map is not found in this list, a strategy must be found to construct the mapping. A plethora of techniques are described by Moretti (149) and Ives (124). Briefly, these include composition of elementary maps, near circle transformations (212), and single-step mappings. Near circle transformations are useful when a geometric object can be transformed to a shape close to a circle. Techniques have been perfected to take this near circle and transform it into a circle. Single step transformations encompass Schwarz-Christoffel transformations, and various other specialized mappings. Davis (45) has considered a generalized Schwarz-Christoffel transformation to treat curved boundaries.

Conformal mapping algorithms impose the distribution of the mesh points along the boundaries, as well as their distribution in the interior of the physical domain, once the mapping and the boundaries of the computational domain have been chosen. There is no flexibility in the distribution of nodes across the physical domain. This is particularly evident when the physical domain contains a long narrow finger: the conformal cells there tend to rapidly dilate upon approaching the finger tip, thereby rendering it unresolved. Such fingers occur, for example, in airfoil cascade grids. It is also impossible to cluster grid points along boundary surfaces to resolve salient features present in the initial conditions, or expected in the course of the flow field calculation. On the other hand, most differential equations transform trivially under a conformal mapping. For instance, the Laplace equation is invariant under such a transformation. This is one of the chief reasons why conformal grids are still quite popular. Potential flow theory is still of interest to fluid dynamicists. With the potential equation being invariant under conformal mapping, the equations in transformed space often have analytical solutions, and in turn they produce answers to the original problem posed in physical space.

Without losing the advantages inherent in conformal transformations, it is common practice to apply separate constant scaling transformations along the two compuational coordinate curve directions. This can either be done as a postprocessing step, or within a direct generation of the grid by solving the modified Laplace equations

$$\begin{aligned} b^2 x_{\xi\xi} + a^2 x_{\eta\eta} &= 0 \\ b^2 y_{\xi\xi} + a^2 y_{\eta\eta} &= 0 \end{aligned} \tag{14}$$

where a and b are respectively the proportionality constants for the ξ and η coordinate curves. More general transformations are obtained by first transforming the computational space coordinates into a new set of coordinates (ξ_1, η_1) by

$$\begin{aligned} \xi_1 &= h_1(\xi) \\ \eta_1 &= h_2(\eta) \end{aligned} \tag{15}$$

where all computational space coordinates lie in the range $[0, 1]$. This is followed by a conformal mapping applied to (ξ_1, η_1). The result is an orthogonal mapping. With the previous system 14, the scaling functions can be nonlinear. Of course, the resulting system of equations has to be solved numerically in most cases, thereby losing the advantage of analytically defining the coordinate transformations. There is a large body of literature on conformal mappings. Most notable are the reviews by Moretti (149) in 1980, and by Ives (124) in 1982. A conference on conformal mapping was also held by Trefethen (231).

Recent contributions to conformal mapping algorithms have been made by Ives (125), Ives and Menor (123), Halsey (107,108) and Baker (14). Ives (125) considers a three-dimensional inlet/centerbody grid. Two directions are gridded with a conformal mapping, while the third direction is the result of independent algebraic stretchings along the two coordinate directions involved in the conformal map. The result is a three-dimensional grid orthogonal in two directions but not in the third.

Baker (14) approaches the three-dimensional problem from a different perspective. At each stage, he applies a nearly conformal mapping for a pair of coordinate variables. By performing a sequence of such mappings for distinct pairs, the final composed map attains some three-dimensional generality. This generality occurs in the same spirit as in the development of Euler angles in classical mechanics. In typical applications, Baker uses the parabolic (Eq. 12) and Joukowski (Eq. 13) transformations in combination with shearing transformations (cf. Eq. 16). While the conformal maps were employed for the gross movement of the bodies into general desired positions, the shearing was used for the final and more modest push into the actual desired positions. For the aircraft mappings, the sheared Joukowski map was used to send the fuselage into a plane. With the wings and tails carried along with this map, we arrive at the next stage. Noting that the basic character of the carried wing is preserved, a sheared parabolic mapping is then used to unwrap the wing that now comes out of the plane. Another stage is thus reached and this process continues until all elements are appropriately treated. The entire composition is then used to generate the grid. This involves a formal inversion.

One cannot, with a conformal transformation, impose both the grid positions and the angle the transverse coordinate lines make with the boundaries. There are also situations where an imposed node distribution is required across the flow field. For example, there might be a need for the inclusion of a small region in which the grid is Cartesian, even though the grid might otherwise be polar. Local controls of this type, and others, cannot be incorporated within the theory of conformal mapping. Even though local controls could be applied to the grid in the computational plane, and the resultant grid subjected to an appropriate conformal mapping, one would not know a priori how the resultant grid in physical space would look. We therefore turn our attention to a class of algorithms which operate in physical space, and can be expressed analytically.

Unidirectional Interpolation

A large class of problems depend on a grid that spans two boundaries, typically a solid object and the far-field. These two geometries includes two- and three-dimensional pipes and windtunnels. They, however, can have a singular curve which must be joined up with the walls. For axisymmetric configurations, this singular curve is a straight line, from which the grid lines should leave orthogonally. Inlets may contain shocks, which further require the grid points to be clustered in localized areas to insure that the flow field can be calculated accurately. Denote by $\vec{r}_1(\vec{\xi})$ and $\vec{r}_2(\vec{\xi})$ two vectors in physical space. As a function of the computational coordinates $\xi_i \in [0,1]$, these vectors trace out a curve in 2-D space when $\vec{\xi} = \xi$ and a surface in three-dimensional space when $\vec{\xi} = (\xi_1, \xi_2)$. Unless otherwise specified, the discussion that follows is restricted to two-dimensional space. Although not always explicitly stated, most of the algebraic techniques that will be discussed are readily extended to higher dimensions. The constructive computational coordinate η, also in the range $[0,1]$ follows the family of coordinate lines that intersect the two boundaries at a non-zero angle.

Shearing transformations. The simplest transformation that connects the boundaries $\vec{r}_1(\xi)$ and $\vec{r}_2(\xi)$ is the shearing transformation

$$\vec{r}(\xi,\eta) = \vec{r}_1(\xi) + \eta(\vec{r}_2(\xi) - \vec{r}_1(\xi)) \tag{16}$$

which describes the migration of the vector $\vec{r}(\xi,\eta)$ from the inner boundary $\vec{r}_1$ when $\eta = 0$ to the outer boundary $\vec{r}_2$ when $\eta = 1$. The coordinate ξ specifies the location of the radius vector $\vec{r}$ along the boundary. In other words, ξ is the boundary parametrization. Equation 16 is the parametric equation of a family of straight lines joining the two boundaries between points of equal ξ. Gridding the physical domain consists of choosing a parametrization of both boundaries, and joining points of equal parameter value with straight lines. In this manner, it is possible to insure orthogonality at one of the boundaries, but not both, unless they are parallel. For the case of symmetric nozzles, the wall is first parametrized. Projection of the wall nodes onto the nozzle axis usually determines the parametrization of the axis. Application of Eq. 16 generates the intermediate $\eta = cst$ coordinate curves. In the shearing transformation defined by Eq. 16, the variable η can be replaced by any increasing strictly monotonic function $\Gamma(\eta)$ which satisfies the two conditions

$$\begin{aligned} \Gamma(0) &= 0 \\ \Gamma(1) &= 1 \end{aligned} \tag{17}$$

A suitable choice of Γ gives the user control of the radial clustering of nodes along the $\xi = cst$ curves. Eiseman (59) has analyzed and applied the shearing transformation in two dimensions. The particular application was the gridding of a periodic cascade of airfoils.

Although full control over node position along the boundaries in physical space has been achieved, it is at the expense of angle specification. On the other hand, under restricted circumstances, the angle can be specified on one boundary and the positions on the other. If multiple coordinate systems are patched together into a larger assembly, shearing transformations can by themselves, produce an overall grid with derivative discontinuities Specifically, slope discontinuities are usually propagated inwards from the boundaries. Several alternatives are available to eliminate the problem. The most straightforward is to smooth the composite grid in a postprocessing step. However, by doing so, one loses some of the fine control. Another solution is to increase the number of free parameters in the coordinate transformations, either by choosing one or more interior nodes or by making use of derivative information along the boundaries. The first approach leads to the multisurface transformation which is the subject of a later section. If the node locations and the tangent vectors for the $\xi = cst$ curves are specified at both boundaries, the resulting mapping is a Hermite transformation.

Hermite transformations. If in addition to boundary coordinates, derivative information is also provided at the boundaries, the previous linear mapping in η must be replaced by a cubic mapping. This allows the specification of four boundary conditions. As a function of the two tangent vectors $\partial\vec{r}_i/\partial\eta$, $i = 1,2$, the global cubic Hermite transformation is given by

$$\begin{aligned}\vec{r}(\xi,\eta) &= (1 - 3\eta^2 + 2\eta^3)\vec{r}_1(\xi) + \eta^2(3 - 2\eta)\vec{r}_2(\xi) \\ &\quad + \eta(1-\eta)^2\frac{\partial\vec{r}_1}{\partial\eta} + \eta^2(\eta - 1)\frac{\partial\vec{r}_2}{\partial\eta}. \qquad (18)\end{aligned}$$

The tangent vector $\partial\vec{r}_1/\partial\eta$ is interpreted as the derivative of $\vec{r}(\xi,\eta)$ with respect to η at $\eta = 0$. An identical statement is true regarding $\vec{r}_2$ at $\eta = 1$. Hermite interpolation forms the backbone of the two-boundary technique developed by Smith (187). Unfortunately there are no guidelines for choosing the magnitudes of the first derivatives of the position vectors with respect to η at the boundaries. If η were the arc length of the coordinate curve, then it is well known, that the tangent vectors would have a magnitude of unity. However, η has no relation to arc length. If the magnitudes of the tangent vector on either boundary is too large, the coordinate curve $\vec{r}(\xi,\eta)$ could backtrack upon itself and become double-valued.

There are various generalizations of the Hermite polynomial. Higher derivatives can be introduced on the boundary, or as previously mentioned, interior nodes can be added to the interior of the domain. In the latter case, it would lead either to a global polynomial with a higher degree, or to local Hermite polynomials. The distinction between between local and global interpolation functions is explained further in the next section on the multisurface transformation which contains the shearing and Hermite transformations as special cases.

The multi-surface transformation. Although the shearing and Hermite transformations described in the two previous sections have greatly extended the amount of grid control made available to the user, this control is still restricted to the boundaries. In an attempt to remove this last restriction, Eiseman developed the multisurface transformation (65,61,63,64) of which the Hermite and shearing transformations are two special cases. The multisurface transformation distinguishes itself from the previous interpolation formulae in two regards. First, intermediate surfaces are introduced between the inner and outer boundaries. For convenience, renumber the boundaries to vary in succession from 1 to N. Under this new reordering, the inner boundary is labeled $\vec{r}_1(\xi)$ and the outer boundary

$\vec{r}_N(\xi)$. The constructive coordinate, as before, is the radial coordinate η. From now on, radial coordinate lines refer to the coordinate lines $\xi = cst$. The second novel feature of the multisurface transformation is that the interpolating curve does not pass through prescribed node locations in physical space (as for the shearing and Hermite transformations). Rather, it is a vector field that is interpolated, in the sense that the interpolating curve is tangent to a prescribed set of tangent vectors at, as yet, undetermined locations.

The $N-2$ intermediate surfaces, each parametrized by ξ, serve to guide the radial coordinate line from the inner to the outer boundary. The simplest generalization to the shearing transformation is to let the radial lines be a succession of line segments spanning successive surfaces at a fixed value of the parameter ξ. However, this would generate a coordinate line discontinuous at the inner surface boundaries. On the other hand, with this discontinuous curve firmly in mind, it is easy to visualize a smooth curve leaving $\vec{r}_1$ along the first tangent vector, tangent to each inner line segment. and arriving at $\vec{r}_N$ along the $(N-1)^{th}$ tangent vector. This is exactly how the intermediate surfaces control the shape of the radial coordinate curve when the interpolation functions are piecewise linear (63).

Let the $N-1$ tangent vectors be defined from successive surfaces by

$$\vec{V}_n(\xi) = A_n(\vec{r}_{n+1}(\xi) - \vec{r}_n(\xi)) \tag{19}$$

where arbitrary lengths A_n are assumed for the moment. As a function of the radial coordinate η, the general tangent vector field is

$$\frac{\partial \vec{r}(\xi,\eta)}{\partial \eta} = \sum_{i=1}^{N-1} \psi_i(\eta)\vec{V}_i(\xi). \tag{20}$$

To insure that the general tangent vector passes through the specified vector field $\vec{V}_i(\xi)$, the interpolation function ψ_i must satisfy the normality conditions

$$\psi_i(\eta_j) = \delta_{ij} \qquad i,j = 1,2,\cdots,N-1 \tag{21}$$

at the $N-1$ values of η_j, which are only subject to the restriction that $0 = \eta_1 < \eta_2 < \cdots < \eta_{N-1} = 1$. This simply means that the tangent vector is equal to the individual vectors $\vec{V}_i$ at monotonically increasing values of η along the radial coordinate curve. After an integration of Eq. 20 with respect to η and a normalization for A_i, the position of $\vec{r}(\xi,\eta)$ is given by

$$\vec{r}(\xi,\eta) = \vec{r}_1(\xi) + \sum_{i=1}^{N-1} \frac{G_i(\eta)}{G_i(1)}(\vec{r}_{i+1}(\xi) - \vec{r}_i(\xi)) \tag{22}$$

where

$$G_i(\eta) = \int_0^\eta \psi_i(\eta')d\eta'. \tag{23}$$

The normalization is required to match the endpoint location $\vec{r}_N(\xi)$ with $\vec{r}(\xi,1)$, and is represented by the $G_i(1)$ in the denominators. Two ingredients are therefore necessary to define the multisurface transformation once a set of multisurfaces have been chosen and parametrized: the interpolants ψ_i and the intermediate values $\eta = \eta_i$ at which the tangent of the radial coordinate curve is equal to the vectors $\vec{V}_i$. It is this wealth of options which is the strength of algebraic methods. Not only is there a wide choice of parameters, but most of them can be interpreted geometrically. Unfortunately, with this wide choice comes the hardship of choosing. However, some simple choices can be made which work most of the time (60). Polynomials that satisfy Eq. 21 are simply the Lagrange polynomials. However other functional forms are possible, although some may be harder to manipulate than others. Note that integrals of ψ_n appear in the expression for $\vec{r}$. Therefore, the derivative order of the interpolant for the coordinate transformation $\vec{r}(\xi)$ is one greater than that of ψ_n. Setting

$$\Gamma_n(\eta) = \frac{G_n(\eta)}{G_n(1)}, \tag{24}$$

Eq. 22 transforms into a format reminiscent of the shearing transformation:

$$\vec{r}(\xi,\eta) = \vec{r}_1(\xi) + \sum_{i=1}^{N-1} \Gamma_i(\eta)(\vec{r}_{i+1}(\xi) - \vec{r}_i(\xi)). \tag{25}$$

Interpolants. For two degrees of freedom ($N = 2$) and accordingly, $\psi(\eta) = cst$, the shearing transformation is retrieved exactly. If $N = 3$, there are two interpolation functions: $\psi_1(\eta) = \eta$ and $\psi_2(\eta) = 1 - \eta$. With three degrees of freedom, one can specify node locations on two boundaries, and coordinate curve angles on one of them, or alternatively, angles on both boundaries, but the node locations on a single one. The more interesting case is when $N = 4$ which corresponds to the 4-surface transformation.

When $N = 4$, the simplest global polynomial interpolants (up to a constant multiple) are

$$\begin{aligned} \psi_1(\eta) &= (\eta - 1)(\eta - \eta_2) \\ \psi_2(\eta) &= \eta(\eta - 1) \\ \psi_3(\eta) &= (\eta - \eta_2)\eta \end{aligned} \tag{26}$$

$$\tag{27}$$

which leads to a cubic transformation in η. When $\eta_2 = 1/2$, an identification can be made between the 4-surface transformation and the Hermite interpolation touched upon earlier. In this case, the tangent vectors $\partial\vec{r}_i(\xi)/\partial\eta, i = 1,2$ in Eq. 18 are related to the positions of the two intermediate surfaces by

$$\begin{aligned} \frac{\partial\vec{r}_1(\xi)}{\partial\eta} &= 6(\vec{r}_2(\xi) - \vec{r}_1(\xi)) \\ \frac{\partial\vec{r}_2(\xi)}{\partial\eta} &= 6(\vec{r}_4(\xi) - \vec{r}_3(\xi)). \end{aligned} \tag{28}$$

The principle advantage of this equivalence is the geometric interpretation afforded by the multisurface transformation. As long as the intermediate surfaces are disjoint from each other, the radial curves are prevented from backtracking on themselves, unless the series of line segments joining points of equal parameter value ξ on the successive surfaces $\vec{r}_k$ overlap on themselves as well. In other words, Eq. 28 places an upper bound on the magnitude of the tangents in the Hermite formula if overlap is to be avoided. This bound is not absolute, however, it could serve as a guideline in codes using Hermite polynomials to reduce some of the arbitrariness in choosing the magnitude of the tangent vectors.

If the number of degrees of freedom is increased, so is the available control over the grid. Once $N \geq 6$, the curvature of the radial coordinate line at the inner and outer body can be specified as well. This controls the distance the radial line extends away from the boundary before it curves away to eventually join up with the outer boundary.

Up until now, the discussion has focussed on global polynomials that spanned the entire region from inner to outer boundary. It is therefore evident that if an intermediate surface, $\vec{r}_i$ is moved, the grid will be affected along the entire length of the radial curves for all values of ξ influenced by the movement of $\vec{r}_i$. Full control over the grid has already been achieved in the ξ direction, since the radial lines are independent of each other. The radial lines have not yet been subdivided into units independent of each other. This can however be done quite simply. If the interpolants are no longer defined over the entire range of η, but instead over a limited range, the shape of an intermediate surface could be modified without affecting entire radial lines. The simplest local interpolants are linear hat functions; ψ_k is one at η_k and zero at the two adjacent radial coordinates η_{k-1} and η_{k+1}. At the boundaries, the interpolants are-one sided (63). These functions are only useful for two-dimensional grid generation which do not require second derivative calculations. In three-dimensions, the lack of second derivative continuity of the vector $\vec{r}$, along the coordinate curve spanning two surfaces, could generate coordinate curves with discontinuous curvature. This is is forbidden. Eiseman has developed the theory of local piecewise quadratic interpolants (64), which generate piecewise cubic radial coordinate lines that are second derivative continuous and can be safely employed in three-dimensions.

Algebraic methods based on global interpolating functions propagate boundary discontinuities into the interior of the domain. With local interpolants, by contrast, this can be limited. More specifically, it can be shown that if the distance from the corner to the $\vec{r}_2(\xi)$ is about half the first grid point spacing, the discontinuity will not be propagated inward (63) if $\vec{r}_2(\xi)$ is smooth. In elliptic solvers, discontinuities arising on the boundaries will not propagate inward because of the inherent smoothing of the differential operator.

Uniformity. The next step in achieving a maximum of degree of control over the grid is to specify an arbitrary distribution of nodes in the radial direction. Upon achievement of a uniform distribution, it can be composed with a suitable radial transformation to generate the desired placement of nodes. Therefore the question arises: under which conditions is it possible to generate a grid with the multisurface transformation, and yet retain a uniform radial distribution. The simplest approach is to postprocess an existing grid. Cutting the radial coordinate curves into equal pieces produces a nearly uniform radial distribution, unless the radial curves deviate strongly from straight lines (69). Therefore, uniformity is defined relative to a straight line of direction $\vec{r}$, upon which the radial nodes are projected. In particular, this line is often taken to pass through $\vec{r}_1$ and $\vec{r}_N$. The cumulative arc length along the line supported by $\vec{r}$ is

$$s(\xi,\eta) = [\vec{r}(\xi,\eta) - \vec{r}_1(\xi)] \cdot \vec{r}(\xi). \tag{29}$$

Uniform distribution along $\vec{r}$ is equivalent to the requirement that $s(\xi,\eta)$ be a linear function of η. This statement translates into projected increments for intermediate segments and imposes

$$\sum_{k=1}^{N-1} \frac{\psi_k(\eta)}{\psi_k(\eta_k)} = 1. \tag{30}$$

which is a condition on the interpolants $\psi_k(\xi)$. Altogether, the number of free parameters in the vectors is reduced by one. Both the global polynomial and the local piecewise linear interpolants automatically satisfy the uniformity condition expressed by Eq. 30. This is not so in general for the piecewise quadratic functions (64); they were explicitly constructed to satisfy it.

Transfinite Interpolation. The various algorithms presented in the previous sections, with the culmination of the multisurface transformation, are all specialized to explicitly generating radial curves leaving an inner boundary $\vec{r}_1(\xi)$ and reaching an outer boundary $\vec{r}_N(\xi)$ However, in many cases, there are 4 boundaries in two-dimensional configurations, and 6 boundaries in 3-D which must be matched by the corresponding coordinate curves or surfaces. Projectors have long been the basis for multidirectional interpolation, and their understanding is greatly simplified with the introduction of Boolean operations. A shearing transformation is the simplest unidirectional form of interpolation, and forms the basis on which Boolean theory is introduced. More details are provided by Gordon (102),Gordon and Hall (103), and Gordon and Thiel (104) with practical implementations given by Smith (187) and Eriksson (77). Eriksson applied the transfinite interpolation concept to the gridding of fuselage/wing combinations on aircraft. He achieved considerable success by incorporating boundary derivative information into the formalism.

Consider the shearing transformation

$$\vec{r}(\xi,\eta) = (1-\eta)\vec{r}_1(\xi) + \eta\vec{r}_2(\xi) \tag{31}$$

between the two boundaries $\vec{r}_1$ and $\vec{r}_2$. Let $\vec{F}(\xi,\eta)$ be a member of the collection of all transformations from the unit square $0 < \xi < 1$ and $0 < \eta < 1$ to the physical domain bounded by any two boundary segments $\vec{r}_A(\xi)$, $\vec{r}_B(\xi)$. Among all these mappings, the shearing transformation between $\vec{r}_A(\xi)$ and $\vec{r}_B(\xi)$ can be retrieved by applying a suitable operator Q_η to $\vec{F}$. The action of Q_η is simply to select the shearing transformation from all of the transformations between a given pair of boundaries. Viewing each transformation as if it were a distinct element of a vector space, the action is intuitively seen as a projection onto the shearing element. By rigorous definition, Q_η is a projection operator since a double application of Q_η to the arbitrary transformation $\vec{F}$ gives the same result. Symbolically, the shearing transformation is

$$Q_\eta[\vec{F}] = (1-\eta)\vec{F}(\xi,0) + \eta\vec{F}(\xi,1). \tag{32}$$

where $\vec{F}(\xi,0)$ and $\vec{F}(\xi,1)$ are respectively the boundaries $\vec{r}(\xi,0)$ and $\vec{r}(\xi,1)$, or equivalently $\vec{r}_1(\xi)$ and $\vec{r}_2(\xi)$. As a function of η the transformation at fixed values of ξ generates straight line segments. Straight lateral boundary segments are often too restrictive. Information about the true lateral boundary shapes is included in the multidirectional coordinate transformation by considering the shearing transformation

$$Q_\xi[\vec{F}] = (1-\xi)\vec{F}(0,\eta) + \xi\vec{F}(1,\eta), \tag{33}$$

where $\vec{F}(0,\eta)$ and $\vec{F}(1,\eta)$ represent the alternate lateral set of physical boundaries $\vec{r}(0,\eta)$ and $\vec{r}(1,\eta)$. Although this example is illustrated with shearing transformations and is two-dimensional, any of the transformations mentioned in the previous section are suitable, and different directions may have different interpolation formulae. Extensions to three-dimensions are straightforward. As explained in (69), the product of Q_ξ and Q_η gives a *tensor product interpolant* which maps the four corners of the computational domain to the four corresponding corners of the physical domain. The remaining boundary points between the two domains have no correspondence under this mapping. This occurs because the boundaries map onto line segments between the corners. By contrast, the simple sum $Q_\xi + Q_\eta$ maps each boundary to the sum of a line segment and the actual physical boundary. By using the tensor product mapping to remove the boundary line segments, we arrive at the Boolean sum

$$Q_\xi \oplus Q_\eta = Q_\xi + Q_\eta - Q_\xi Q_\eta. \tag{34}$$

The action on $\vec{F}$ then clearly transforms computational boundaries to their respective counterparts in the physical plane. In practice, the previous equation is broken up into several components. For example in three dimensions one would write:

$$\vec{F}_1 = Q_\varsigma[\vec{F}], \tag{35}$$

$$\vec{F}_2 = \vec{F}_1 + Q_\eta[\vec{F} - \vec{F}_1], \tag{36}$$

$$\vec{F}_3 = \vec{F}_2 + Q_\xi[\vec{F} - \vec{F}_2], \tag{37}$$

where ς is the computational variable in the third direction. The previous transformation is symbolically represented by $(Q_\xi \oplus Q_\eta \oplus Q_\varsigma)[\vec{F}]$. Boolean maps also find applications in situations where only edges must be mapped in three dimensions, or when only corner correspondence is desired. Once again, it is emphasized that the interpolation functions may differ from one another in independent directions. Further control of the coordinate curve directions is discussed in Gordon and Thiel (104) where points interior to the physical domain become part of the interpolation process. Additional control is available in the form of a control net for the transfinite multisurface transformation given by Eiseman (73).

An interesting vehicle for transfinite interpolation, are the homotopy methods developed by Barger (16) while generating grids over aircraft wing-fuselages combinations.

Briefly, homotopy methods are mappings of a single variable ξ which smoothly transforms an inner surface to an outer surface as the ξ varies from 0 to 1. They are formally univariate interpolation methods, however the formalism allows a more general class of interpolation functions to be used. As a particular example, Barger rewrites the shearing transformation as

$$\vec{r}(\xi,\eta) = A(\xi)[(1-\xi)\vec{r}_1(\eta) + \xi\vec{r}_2(\eta)] \tag{38}$$

where the function $A(\xi)$ is a positive, but not necessarily monotonic, function of ξ, and controls the rate at which the boundary $\vec{r}_1$ changes into the boundary shape represented by $\vec{r}_2$. Barger further extended his formalism to multiple dimensions. Explicit functions can also be constructed between successive pairs of surfaces while maintaining a smooth junction at the interfaces.

Grid Blending. After studying some of the properties,unidirectional interpolation, followed by the mechanisms by which they can be combined to generate multidimensional grids that match all the boundaries, one is sometimes left with an assembly of grids that must be patched together. Some of the techniques are block-structured algorithms, overlaid grids (205,19), and zonal grids (169). However it is sometimes possible to blend independently created grids into a single grid. Steinhoff (208) has developed such an approach which he has applied to the gridding of fighter aircraft (209). A special case of the general blending function described by Steinhoff (208) is the shear transformation when only two grids must be blended together. Assume that grids $\vec{G}_1(\xi,\eta)$ and $\vec{G}_2(\xi,\eta)$ are two grids that have been generated by independent methods. A typical example are the grids around an airplane wing and its fuselage. One requirement of this method is that all the grids have the same number of nodes in each of the computational directions. The composite grid $G(\xi,\eta)$ becomes

$$G(\xi,\eta) = f(\xi,\eta)G_1(\xi,\eta) + (1-f(\xi,\eta))G_2(\xi,\eta). \tag{39}$$

It is clear that the art lies in the choice of positive weight function $f(\xi,\eta)$. Near the replication of grid 1, this weight should be close to unity. At the other extreme, the weight function should decrease towards zero as the second grid is approached. Steinhoff presents several examples, among which is a composite blended grid around a car.

Postprocessing. Once a grid or a series of grids have been generated about a physical configuration, it may be desirable to locally modify their structure, smooth them, or blend them together into an overall grid. Eiseman (71) has developed a postprocessing step that allows the user to concentrate nodes at a point, along a coordinate line, or in a region. The procedure is akin to local movement of the coordinate lines towards the area to resolve, at the expense of surrounding regions. Depletion of nodes occurs in local regions around the area to be resolved. Clustering occurs in a smooth manner, that is to say that a smooth coordinate mapping remains smooth.

Another approach to postprocessing a grid is relevant to the partial differential equation approach. The idea is to generate an approximate grid by any one of a variety of methods, and then smooth it using an elliptic solver. When the known grid point locations are substituted into the left-hand side of a Laplace equation as a function of the computational coordinates (ξ, η), a forcing function is obtained for the right-hand side. After smoothing out the high frequency waves from the forcing function, a new grid is computed as the solution to the resultant differential equation. This has been reported by Thompson (222). Kennon and Dulikravich (129,128) prefer to smooth the grid using a variational procedure where orthogonality can be enhanced. In the end, virtually any elliptic method can be employed as a filter. The amount of extra work involved will, of course, grow with the demands. In fact, algebraically generated grids are often the initial conditions for the elliptic methods discussed below.

PARTIAL DIFFERENTIAL EQUATION METHODS

When the grid is the numerical solution to a system of partial differential equations, the controls are determined by the system definitions. Of the various types of defining systems, those which are elliptic contain the strongest control properties; albeit, they are not as fast as the hyperbolic and parabolic methods that can generate grids in only one marching pass through a region. The hyperbolic methods are orthogonal and are determined by specifications over the field that vary from an algebraic metric relationship, to a given family of curves, and finally to prescribed cell volumes from a reference grid. Transfinite interpolation with shearings is also noted to come from a hyperbolic system, although it is a non-orthogonal, non-marchable explicit algebraic transformation. Altogether, the hyperbolic method provides a quick means to get a decent grid from somewhat modest field specifications. The speed issue is somewhat tempered, however, by the availability of fast numerical algorithms and fast computers. As a consequence, the ability to control the outcome is the primary concern; and thus we focus our attention upon elliptic partial differential equations which are more simply called *elliptic methods*. There, we start by considering the Laplace system of Winslow and then progress into the Poisson system of Thompson, Thames, and Mastin where controlling terms are first inserted. To understand the basic controls, we examine their construction, effect, and application. This includes basic clustering mechanisms, inward propagation of boundary node distributions, specification of angles at boundaries, and the effect of boundary curvature on the inward distribution of points from that boundary. A natural framework for elliptic methods is provided by variational statements. In such statements, the controls are easily established for distinct purposes, but the actual execution is much more complex.

Hyperbolic Methods

When a grid must be generated about a single body, hyperbolic methods can be employed. From a given grid on the body, the generation process proceeds by spatially evolving the grid in marching steps that progressively depart from the body. The extent of coverage for the surrounding field is determined by the number of steps, the size of each step, and the defining metric relationship of the hyperbolic system. While the choices concerning steps are a common element of any spatially marching scheme, the choice of defining system provides the primary separation of the various methods.

A Specified Relationship Between Metric Elements. The common part of each defining system has been the assumption of some form of orthogonality. In two-dimensional Euclidean space, orthogonality is given by the metric equation

$$x_\xi x_\eta + y_\xi y_\eta = g_{12} = 0. \tag{40}$$

This equation is split into two equations by Starius(198) who considers the Cauchy-Riemann form

$$\begin{aligned} x_\eta &= -F y_\xi \\ y_\eta &= F x_\xi \end{aligned} \tag{41}$$

that exactly satisfies it. By a direct substitution into Eq. 3, the function is easily found to be

$$F = \sqrt{\frac{g_{22}}{g_{11}}}. \tag{42}$$

While the general format of Eqs. 41 - 42 has often been used by others for orthogonal grid generation (65), the distinguishing feature from Starius (199) is the choice of F. This choice is accomplished in such a way as to assure that the system of equations 41 is hyperbolic and therefore can be spatially marched. Altogether, his rigorous selection criteria seems to have directed him to a somewhat conservative choice. This is evident from the fact that the generation process must be stopped within a certain distance from concave portions of the body or else a grid overlap would occur because of a singularity in the mesh. The selected from of F, however, does have the advantage of simplicity. It is given as a low order constant coefficient rational function of g_{11}. With the defined function, the generating system of Eqs. 41 is completely established over the entire field.

While the algebraic construction of F provided Starius with a direct global definition, it also presented the rigidity that limited the domain of applicability to small bands near the body surface. This lead to some developments with overlapping grids (196,197,198,133) so that the small bands could be employed in a more practical context. By relaxing the above rigidity, it is natural to consider definitions that are distributed over the field with some degree of flexibility.

Orthogonal Trajectories Through a Specified Family of Curves. The first and most basic way to achieve an acceptable level of flexibility is to consider orthogonal trajectory methods. The specified properties over the entire field come in the form of one family of surfaces that can qualify as coordinate surfaces and that conform to the body on one side and to a given outer (inner) boundary on the other. With the ready capability to generate nonorthogonal grids in a variety of ways, the specified family is most directly obtained by extracting one family of coordinate curves from a given transformation. To accomplish this task efficiently, and to provide for the possibility of arbitrarily fine resolution of the trajectory paths, the nonorthogonal transformation should be algebraic because of its explicit and continuous definition. The simplest, and most commonly used transformation is the direct shearing (Eq. 16) between the two opposing boundaries. Should lateral boundaries also be present and be preserved by the trajectories, then a Hermite transformation must be employed as a transverse Boolean summand to supply orthogonality at those boundaries. To add a high degree of flexibility, the Boolean sum of shearing and Hermite transformations should be replaced by multisurface transformations (Eq. 22) with local interpolants.

In two-dimensional Euclidean space, the given nonorthogonal transformations typically appear in the form

$$\begin{aligned} x &= x(\alpha, \beta) \\ y &= y(\alpha, \beta) \end{aligned} \tag{43}$$

where curvilinear coordinates (α, β) are directly sent into Cartesian positions $\vec{r} = (x, y)$. To compute the metric coefficients g_{11}, g_{12} and g_{22}, we use α and β in place of ξ and η in Eq. 3. Here, we note that because of the nonorthogonality, g_{12} does not vanish. The fact that one family of coordinate curves is to be preserved in the anticipated orthogonal system means that a second transformation for composition with the first must assume a special form. Such a form is given by

$$\begin{aligned} \alpha &= \alpha(\xi, \eta) \\ \beta &= \beta(\eta) \end{aligned} \tag{44}$$

which is the general expression of a transformation where coordinate curves with constant η become coordinate curves with constant β. The general dependence of β upon η serves to indicate the possibility of redistributing the coordinate curves within the given family: without redistribution, the dependence would simply be $\beta = \eta$. Each coordinate curve with fixed η is then to get a new distribution of points along it. This translates into a redistribution for the coordinate α for each fixed η; that is, a distribution function with respect to the new curvilinear variables ξ along each such curve. Altogether, we arrive at the desired transformation of Eq. 44 whose composition with the nonorthogonal system of Eq. 43 then preserves the curves of constant β.

The next task is to choose the successive reparameterizations with respect to ξ in such a way as to produce an orthogonal coordinate system in the physical space of (x, y). The basic orthogonality condition of Eq. 40 is employed for this purpose. Under the action of chain rule operations and with the assumption of nonsingularity, the condition becomes a hyperbolic partial differential equation for ξ and is given by

$$\frac{\partial \xi}{\partial \beta} - \frac{g_{12}}{g_{11}} \frac{\partial \xi}{\partial \alpha} = 0 \tag{45}$$

where the metric coefficients are those that come from the nonorthogonal coordinates as discussed immediately after Eq. 43. The characteristic directions for this hyperbolic equation are given by

$$\frac{d\alpha}{d\beta} = -\frac{g_{12}}{g_{22}}. \tag{46}$$

Since a constant value of ξ appears along each characteristic curve generated by Eq. 46 and since such curves are mapped into (x, y) by Eq. 43, we get the remaining family of coordinate curves for the orthogonal transformation. When a distribution of points is prescribed along an initial coordinate curve with fixed β in the given nonorthogonal system of Eq. 43, a corresponding distribution appears in α and in ξ in such a way that the forward composition reproduces the prescribed distribution. The process of solving for the characteristic curves from Eq. 46 is just the process of computing orthogonal trajectories from each prescribed point on the initial curve. As [illegible]equence, the initial distribution of points is propagated across the field: the only possible means for redirection would then have to come from the choice for nonorthogonal transformation in Eq. 43.

In a natural way, the orthogonal trajectories have been developed as the images of characteristic curves under the given nonorthogonal transformation. In the generalization to two-dimensional surfaces, this rigid adherence to the composition of mappings represents an advantage since the surface geometry would not be altered in the technical process of generating the trajectories. This general development is presented by Eiseman (65) in his review of orthogonal grid generation.

While the advantage of adhering to the composition has been noted, a somewhat natural inclination is to proceed with the construction directly, particularly when it is being done in two-dimensional Euclidean space. This is simply the framework which is most familiar to the typical scientist. To bring the governing hyperbolic system into this context, we consider the simple example of generating coordinates from the x-axis up to a prescribed positive function $y = f(x)$. The immediate simple choice for the nonorthogonal transformation of Eq. 43 is to set

$$\begin{aligned} x &= \alpha \\ y &= \beta f(\alpha). \end{aligned} \tag{47}$$

The characteristic directions (Eq. 46) then become

$$\frac{d\alpha}{d\beta} = \frac{\beta f(\alpha) f'(\alpha)}{1 + [\beta f'(\alpha)]^2} \tag{48}$$

and upon using the chain rule to get $d\alpha$ and $d\beta$ in terms of dx and dy we arrive at the familiar constructive form

$$\frac{dy}{dx} = -\frac{1}{\beta f'(x)} \tag{49}$$

of negative reciprocal slope.

Regardless of the numerical methods employed, the essential ingredients of orthogonal trajectory strategies are the selection of a nonorthogonal transformation and the initial conditions for the trajectories. The numerical methods then serve only to gain an accurate approximation of the analytical formulation which is fully expressed by the hyperbolic equation 45.

With the selection of a shearing transformation from a body to a surrounding far field boundary, the orthogonal trajectories are seen to leave concave body sections with a fair amount of compression. The consequence, is that trajectory clusters are propagated into the far field from each concave body section even if the pointwise distribution along the entire body is uniform. The intensity of the propagated clusters depends mostly upon the severity of the concavity.

Specified Cell Volumes from a Reference Grid. To address the concavity problems, either the nonorthogonal transformation or the hyperbolic equation must be changed. With the latter consideration, Steger and Sorenson (203) and Steger and Chaussee (204) were lead to prescribe the field values from a model reference grid. This format represents an increase in flexibility beyond the previous use of nonorthogonal transformations which represented a similar increase beyond the rigid algebraic statements of Starius (198). The reason is that a reference grid does not have to conform with a body geometry. Its only use is to define a desired metric property over the field in such a way that the designer has a ready intuitive picture of the global prescription. With a polar type (O-type) grid about a body, the natural choice is to form an annular system where the body is referenced by an inner circle with the same total arc length as the body. When the pointwise distribution over the body is uniform, the reference grid is simply a polar coordinate system over the annular region. While cell diagonals were considered, a more basic metric quantity is the cell volume. As a consequence, the primary system that Steger considered was given by orthogonality (Eq. 40) and volume specification (J of Eq. 4). Altogether, this is given by

$$\begin{aligned} x_\xi x_\eta + y_\xi y_\eta &= 0 \\ x_\xi y_\eta - x_\eta y_\xi &= V(\xi,\eta) \end{aligned} \tag{50}$$

where $V(\xi,\eta)$ is the specified Jacobian which in finite terms comes from the cell areas of the reference grid. The system of Eq. 50 was shown by Steger to be hyperbolic, and therefore, to permit a solution by spatially marching away from the body. By using the polar reference grids, the previous compression problem inherent in the orthogonal trajectory methods was overcome as the orthogonal grid progressively adjusted to the cell volumes from the reference grid. In fact, upon approaching the far field, the grid so generated converged towards a polar system. The actual outer boundary, however is unspecified. In a graphical intuitive sense, the cells would stretch in a sufficiently rapid manner from concave boundary segments to permit a safe escape from the concavity problem which otherwise would have caused unretrievable compression as in the orthogonal trajectory case or a coordinate singularity as in the case of Starius.

The numerical solution of the hyperbolic system of Eq. 50 is obtained from a direct linearization about the previous coordinate curve. The resultant 2 × 2 block tridiagonal system for the current coordinate curve also contains an artificial viscosity contribution to provide numerical stability and to smear out possible shock-like behavior in the grid. Such behavior could occur directly within the system or propagate into the grid from the body. The typical concern, here, is the propagation of slope discontinuities from the body.

In the extension to three-dimensions, the coordinate curves proceed as before to leave the body orthogonally under the condition of specified cell volumes. Since the orthogonality is with respect to surfaces, there are two orthogonality equations. Altogether, the hyperbolic system is expressed in metric form as

$$\begin{aligned} g_{13} &= 0 \\ g_{23} &= 0 \\ J &= V(\xi,\eta,\varsigma) \end{aligned} \tag{51}$$

where $J^2 = g$ and the third direction ς is the marching direction away from the body.

Because of the flexible arrangement of data specification for a reasonably well-structured grid about a single body, these hyperbolic methods are somewhat ideally suited for applications in the context of overlapping grids whereby each body gets an independently generated grid. While the particular attribute for not rigidly requiring the specification of an outer boundary is an advantage in the chimera scheme (205,18,19), the inability to readily specify it is often disadvantageous. This is the case where, for example, the grid must conform to two opposing boundaries that represent solid bodies.

Transfinite Interpolation Revisited. With more than one solid boundary there are still further hyperbolic systems, although the previous marching element is then gone. A particular example (25) is given by the system

$$\frac{\partial^4 \vec{r}}{\partial\xi^2\partial\eta^2} = 0 \tag{52}$$

with boundary conditions

$$\begin{aligned} \vec{r}(\xi,0) &= \vec{a}(\xi) \\ \vec{r}(\xi,1) &= \vec{b}(\xi) \\ \vec{r}(0,\eta) &= \vec{c}(\eta) \\ \vec{r}(1,\eta) &= \vec{d}(\eta). \end{aligned} \tag{53}$$

The solution is just the transfinite interpolation arising from a Boolean sum of shearing transformations (Eq. 34). Upon reflection, the propagation of boundary slope discontinuities into the field is readily apparent with this transfinite interpolation and can then be interpreted as a hyperbolic effect.

Elliptic Methods

The elliptic methods have stronger control properties than the hyperbolic methods. These properties first appear on a fundamental basis where boundary conditions occur over the entire boundary rather than on only one isolated object. In addition, the basic solution is smooth in the sense of derivative continuity. The smoothness actually occurs in an even stronger sense because the basic frame of reference for most elliptic methods is provided by a Laplacian. In two dimensions, the Laplace equations with Cauchy-Riemann type boundary conditions can be used to determine conformal mappings. See, for example, Chakravarthy and Anderson (35) and Seidl and Klose (183). This arises when the Laplace system is formulated for either physical or curvilinear variables, provided that the resulting transformation is nonsingular. Because of the nonsingular requirement, the latter formulation is preferred. When more general boundary conditions are applied, the conformality is usually lost in a progressive sense: the coordinates are nonorthogonal at the boundaries and become more and more conformal as we leave the boundaries. In terms of a grid mapping, a conformal transformation is explicitly observed as a mapping which sends a uniform distribution of square cells in (ξ, η) into nearly square cells in (x, y) that can have varying sizes. This was analytically stated in Eq. 11. The progression of cells from the boundary are then seen to become closer to a square shape as the region center is approached in the general situation.

Relative to this conformal basis, various controls have been developed to provide for desired distributions and for desired metric structures. The basic control of distributions is the process of clustering points, curves, or surfaces. The basic control of metric structures is typically the enforcement of grid orthogonality to retrieve some of the lost conformality at or near the boundaries. In two dimensions, orthogonality is given only by the metric condition $g_{12} = 0$ while for conformality, the further metric condition $g_{11} = g_{22}$ is also required. The same sort of controls and relative frame of reference also hold in three dimensions and on surfaces. In a fairly general spirit, the fundamental properties of many elliptic methods can be readily seen in a variational setting. This will also be examined.

The Laplace system. With a view towards the competition between differential equations that pull towards conformality and boundary conditions that pull away from conformality, we consider the Laplace system

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= 0 \\ \eta_{xx} + \eta_{yy} &= 0 \end{aligned} \tag{54}$$

with prescribed pointwise distributions on part or all of the boundary. Although conformality could also have been represented by Laplace equations in the opposite direction, the choice of direction and hence dependent variables (ξ, η) is important. While maximum principles are available for either direction, the maximum principle for the choice in Eq. 54 provides a basis for the separation of distinct coordinate curves within a given family of coordinate curves. By contrast, the maximum principle for the Laplace equations for the x and y can only separate the values in the x and y directions of the physical space in which the region of interest is prescribed. Since the x and y directions are not attached to the coordinate curves, the associated maximum principle does not provide the correct force to effectively prevent the curves from overlapping.

Given the more natural inclination to separate entire coordinate curves, we shall examine this mechanism in an intuitive spirit. Since the separation arises in a decoupled fashion, it is sufficient to consider only the curves of constant ξ and the Laplace equation for ξ in Eq. 54. As such, attention is focussed upon a thin strip between two closely spaced coordinate curves in the variable η. In the space of curvilinear variables (ξ, η), the corresponding strip is clearly a thin rectangle bounded by the two verticals established from two successive values of ξ. These values are respectively denoted by ξ_- and ξ_+. Assuming the full range for η, the top and bottom boundaries of the rectangle are horizontal lines defined by the minimum and maximum values for η. To simplify matters, we suppose that the two given coordinate curves do not overlap. The thin strip between them then always has some thickness. By continuity the thickness can be shrunk slightly to provide for a slight expansion later. Now we consider a coordinate curve that corresponds to a fixed value of ξ which is greater than ξ_+. Because of the maximum principle, it cannot overlap the strip or else there would be an internal maximum: the largest strip boundary value ξ_+ would be exceeded. If it just touches the strip boundary, then we use the provided slight expansion to move the boundary. The same argument then means that it cannot intersect the strip. Likewise, no coordinate curve with a fixed value of ξ that is less than ξ_- can intersect the strip. Having seen that coordinate curves on either side of the strip cannot intersect it, we then imagine the situation where ξ_- and ξ_+ gradually approach each other and the strip then converges towards a single coordinate curve with a fixed value of ξ. The limiting case is just the general condition for a nonoverlapping system of ξ coordinates. The corresponding case for the constant η coordinate curves comes from the Laplacian for η and the associated maximum principle and follows the same arguments. A more rigorous discussion involves winding numbers or degree theory and is given by Mastin and Thompson (138) and by Sritharan and Smith (195). By extending the Laplace system of Eq. 54 into three-dimensions with additional second derivative terms in z and with an additional Laplace equation to cover the third curvilinear variable, the same intuitive discussion can be repeated. A rigorous development is given as before by Mastin and Thompson (139).

In a unified manner, the competition between the field equations and the boundary conditions can be stated as a variational problem. In two-dimensions, our objective is simply to minimize the amount by which the Cauchy-Riemann equations fail to be satisfied. At a point in the field, each of the Cauchy-Riemann equations has a residual that in the general circumstance deviates from zero. A measure of nonsatisfaction is clearly given by the sum of squared residuals. The smallest loss over the whole field is then obtained when the integral

$$I_c = \int [(\xi_x - \eta_y)^2 + (\xi_y + \eta_x)^2] dx dy \tag{55}$$

is minimized. In terms of the metric, an equivalent form is obtained when the integrand is replaced by $g^{11} + g^{22}$. If the Cauchy-Riemann conditions are satisfied both in the field and on its boundary, then the integral in Eq. 55 vanishes and the coordinate transformation is conformal. However, if different boundary conditions are applied, then the integral of Eq. 55 contains positive contributions at the boundaries that cannot be removed. The more arbitrarily chosen boundary conditions simply cause the system to be overdetermined. Our only hope for attracting to conformal conditions is then to minimize the integral. When this is done, the unremovable positive contributions at the boundaries will propagate inward from the boundaries because of continuity but will also decay as the field is progressively entered. The corresponding coordinate transformation will then be generally nonorthogonal at the boundaries and will gradually become closer to conformal as the boundaries become more distant. The Euler equations which arise from this minimization problem simply comprise the Laplace system of Eq. 54. Using the metric form of the integrand, the extension into three-dimensions is obvious. The variational problem is the minimization of

$$I_c = \int (g^{11} + g^{22} + g^{33})\,dx\,dy\,dz \tag{56}$$

and the Euler equations are just the direct extension of the Laplace system of Eq. 54 into three-dimensions. As in 2-D, conformality in 3-D is given by vanishing cross metrics ($g_{12} = g_{13} = g_{23} = 0$) and equal diagonal entries ($g_{11} = g_{22} = g_{33}$).

To solve the Laplace system of Eq. 54 or its three-dimensional extension, the independent and dependent variables must be interchanged or else the procedure would become encumbered by additional complexity. Since the grid points in the curvilinear variables (ξ, η) are already known and are simply prescribed to be in a uniform Cartesian format, the most efficient route is to directly solve for the unknown locations in physical space variables (x, y) that correspond to the respective points on the known Cartesian grid. The only known points in physical space are on the prescribed boundaries that typically are directly inserted in the form of Dirichlet boundary conditions. The interchange of variables is just an application of the inverse coordinate transformation which is formally available for derivative operators. With only curvilinear derivatives in the coefficients, the chain rule is used to get the operator relationships

$$\begin{pmatrix} \partial/\partial\xi \\ \partial/\partial\eta \end{pmatrix} = \begin{pmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{pmatrix} \begin{pmatrix} \partial/\partial x \\ \partial/\partial y \end{pmatrix} \tag{57}$$

To transform the given Laplace system, the x and y operators are needed and are given by the formal matrix inverse which yields

$$\begin{aligned} \frac{\partial}{\partial x} &= \frac{1}{J}\left[y_\eta \frac{\partial}{\partial\xi} - y_\xi \frac{\partial}{\partial\eta}\right] \\ \frac{\partial}{\partial y} &= \frac{1}{J}\left[-x_\eta \frac{\partial}{\partial\xi} + x_\xi \frac{\partial}{\partial\eta}\right] \end{aligned} \tag{58}$$

where

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{59}$$

is the Jacobian. The sum of the double application of each yields the Laplace operator and hence the equations with the desired interchange of variables.

While the same development can also be done for the three-dimensional system and for systems on a surface, the algebraic complexity required for each specific derivation is somewhat excessive. A more unified and algebraically succinct approach is to employ the methods of tensor analysis (225,235). The derivation produces the previous results for each case and yields expressions where the coefficients have a natural geometric interpretation. If the curvilinear variables ξ and η are represented by ξ_i for $i = 1$ and 2 and if the physical space variables x and y are represented by x_j for $j = 1$ and 2, then the indices and the metric forms of Eq. 3 are assumed. The three-dimensional case is then obtained with only another index and a corresponding form for Eq. 3 derived from a three-dimensional arc length computation. With the convention that the same lower and upper indices in a product are to be summed over the pertinent index range, the inverted Laplace system is given by

$$D\vec{r} = 0 \tag{60}$$

where

$$D = g^{ij}\frac{\partial^2}{\partial \xi_i \partial \xi_j} \tag{61}$$

and $\vec{r}$ is the position vector in physical space. For the two-dimensional case of Eq. 54, the vector is $\vec{r} = (x, y)^T$. For three-dimensions a third component z is added to $\vec{r}$ while for surfaces the positions become the original surface coordinates. In the case with surfaces, the surface normal vectors also change from point to point, and therefore, their rate of change must be measured. This enters when derivatives are taken and results in

$$D\vec{r} = K\hat{N} \tag{62}$$

where $\hat{N}$ is a unit normal to the surface and K is the mean curvature which gives the rate of change. When the surface is a plane, the curvature term vanishes and the equation specializes to that of Eq. 60.

In the developments with Laplace systems, the first widely disseminated reporting was that of Winslow (245), although there were earlier studies. The earlier works leading up to Winslow's contribution are well documented in the review by Thompson, Warsi, and Mastin (215). The next major step was taken by Thompson, Thames and Mastin (218) who recognized the importance of the method and who extended it to multiconnected regions in two-dimensions. Unlike Winslow who relegated the method to an appendix, they emphasized its application as a general tool. The variational form was first reported in the Cauchy-Riemann form of Eq. 55 by Yanenko et. al. (246) and then several years later in the equivalent metric form by Brackbill and Saltzman (30). In both variational studies, the main emphasis was on solution adaptivity and the conformal part of the integral served primarily as a smooth basis relative to which clustering was to be performed. The general development with surfaces was initiated by Thomas (214), continued by Garon and Camarero (94), and was subsequently developed more thoroughly by Warsi (236,238) who employed the Gauss equations for second derivatives. In both surface studies, the more general Poisson form was considered. A variational approach for surfaces was presented by Saltzman (177).

The Poisson System. While the Laplace system produces a closeness to conformal conditions, and thereby, a desirable degree of smoothness, it also has some drawbacks. The most graphically visible problem is the lack of control over the distribution of points and over the angles of coordinate intersection. In regions near body convexity, the somewhat parallel family of coordinate curves or surfaces is automatically clustered towards the body.

The opposite is true near body concavity. Although this conformal effect may be desired in some circumstances, it is often undesirable. A clear example arises when high resolution is required in a region near fairly high body concavity: the coordinate spacing leaving the body is too rapid because of the conformal repulsive effect. The same repulsive effect also occurs when branch cuts or patch interfaces leave a blunt body or when a singular type of point or curve appears from the choice of cuts or patches. In each of these circumstances, a highly concave region is artificially inserted and thereby inherits the same problem. Moreover, there is often a need to accurately prescribe the spacing as above and to also give the angle from the pertinent boundary. This arises when close spacing and orthogonality are demanded at a body or when a smooth transition is required through a prescribed interface between grids.

To overcome the limitations inherent in the Laplace system and yet at the same time to adhere to the desired conformal smoothness, the next natural step is to consider a Poisson system. In two-dimensions, it appears in the form

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= P(\xi, \eta) \\ \eta_{xx} + \eta_{yy} &= Q(\xi, \eta) \end{aligned} \tag{63}$$

where P and Q are prescribed functions that are employed to control the grid. In a formal sense, these functions force the coordinate curves to move relative to the uniform conditions established by the earlier Laplace system of Eq. 54. As in the earlier discussion where the maximum principle indicated a separation of the coordinate curves for each family, the forces are also applied in a decoupled manner to control that separation. Thus, a discussion of only one of the Poisson equations is all that is needed in order to witness the basic effect of a forcing term. For the curves of constant ξ, the term P in the first equation can be used to move the curves to either side of what otherwise would have been a curve from the harmonic condition of satisfying Laplace equations. In the order of increasing ξ, negative P values cause ξ to fall below the otherwise harmonic ξ, and thus, the associated curve to fall below the otherwise harmonically determined curve. Negative P values then force the function to be subharmonic with an intensity that increases with its magnitude. Similarly, positive P values push curves to the other side in the same manner and represent a superharmonic force. Likewise, Q provides the same control for the constant η curves. General facts on the nature of sub- and super- harmonic functions can be found in Kellogg (127), Epstein (75), and Weinberger and Protter (165).

With the controls decoupled in the sense that the respective actions are applied to corresponding families of coordinate curves, it is reasonable to examine the basic structure of a control for clustering in the context of a simpler one-dimensional Poisson system. This was done in the review by Eiseman (69) where the clustering controls of Thompson, Thames, and Mastin (218) were simply seen to arise quite naturally as controls on the concavity of the coordinate transformation as seen graphically. The basic action there is relative to an equally spaced grid of points.

With a direct extrapolation into two-dimensions, the controls that are simply witnessed in one dimension become controls for curves rather than just points. To cluster curves of constant ξ about a curve for $\xi = \xi_0$, a term of the form

$$-a\frac{\xi - \xi_0}{|\xi - \xi_0|}e^{-b|\xi-\xi_0|} \tag{64}$$

is inserted as a summand within P of Eq. 63. The constants a and b are positive. They respectively represent the clustering intensity and its rate of decay as values of ξ depart from ξ_0. It is important to emphasize here that this attraction is relative only to the curve $\xi = \xi_0$. The intensity decay of the attraction serves to localize the action on both sides of ξ_0.

While only a uniform attraction has been considered, a slight modification will lead to the nonuniform attraction of curves for a cluster about a point on a curve. With the constant ξ curves about $\xi = \xi_0$, the clustering about a point for $\eta = \eta_0$ is accomplished by decaying the intensity also as we leave $\eta = \eta_0$ along the curve. The inclusion of the term

$$-a\frac{\xi - \xi_0}{|\xi - \xi_0|}e^{(-b\sqrt{(\xi-\xi_0)^2+(\eta-\eta_0)^2})} \tag{65}$$

in P is readily seen to meet this objective. For the constant η curves, the same type of terms are employed. In continuation, these clustering terms are readily extended into three-dimensions where now the application is about points, curves, and surfaces.

With the above exponential controls of Thompson, Thames and Mastin (218), Steger and Sorenson (202) considered the iterative determination of the various constants to provide for specified boundary orthogonality and spacing from the boundary. Both the orthogonality and the spacing were constant requirements over an entire boundary with specified pointwise distributions. Accordingly, the two constants in the associated exponential term were found so that the above two constants were matched. For the orthogonality control, the parallel forces to a boundary with fixed points geometrically appear as local rotational forces. A generalization that will allow for a distributed specification of spacings has also been developed and is reported by Thompson, Warsi, and Mastin (225). The original approach of Steger and Sorenson has also been cast into a computer program called GRAPE by Sorenson (192). Yet another strategy is presented by Visbal and Knight (233).

Returning to our earlier observation that harmonic coordinates were attracted to body convexity and repelled by body concavity, the above capability represents a viable strategy to overcome the problem, at least for Dirichlet boundary conditions. A further strategy that is noniterative is to propagate the known boundary distributions into the field. For a body that intersects transverse boundaries, the specified transverse boundary distribution is then continued over the body. More generally, clusters that occur anywhere on a boundary can be sent inward. The first formal reporting of the method was due to Middlecoff and Thomas (143) although some earlier consideration was given by Warsi and Thompson (234). The basic procedure is to derive forcing functions on the boundary from data thereon and to then interpolate into the field to get globally defined forcing functions. In the interpolation care must be taken to include the curvature effects and to perform the process for the entire forcing term rather than treating the curvature part separately. This process is quite detailed and is well documented by Thompson, Warsi, and Mastin (225).

On a more fundamental basis, we can view the controls in the Poisson system as being the maximum allowable under the constraints imposed by the space in which the grid is being generated. In the planar case of Eq. 63, the Gaussian curvature vanishes and provides a relationship between the three metric coefficients that yield general arc length measurements relative to the grid. A knowledge of the three metric coefficients is actually equivalent to knowing the grid. This means that we may specify two of them, solve for the third, and get the grid. Moreover, we may specify two combinations of them and also be completely determined. In a further progression, we may specify up to two metric controlling relationships and still be completely determined. Such relationships can be defined by prescribing the P and Q in Eq. 63. In summary, in two dimensions, there are at most two degrees of freedom available for specification and the Poisson system under consideration consumes both of them. Stated in another way, any grid can be generated by the Poisson system, at least in theory. That is, given a grid, we may directly solve for P and Q to get the appropriate Poisson system. Forgetting the given grid, we can then retrieve it from the P and Q. In a similar spirit, we see that the three-dimensional Poisson system consumes all three possible degrees of freedom. As a consequence, the Poisson system provides a general format or template in which to work. Its general distinction over other equivalent formulations appears in the location where control is inserted. As such, the controls should be viewed as being relative to the harmonic case or else there is no real rationale for this format. Thus, care should be taken when large deviations are considered for the locational advantage might otherwise be lost. In addition, because of the number of available degrees of freedom, the general forcing term expansion proposed by Warsi (236) seems to be of only modest utility: there are more pieces than degrees of freedom.

In the application of the Poisson system, the interchange of dependent and independent variables must be accomplished for the same reasons as in the harmonic case. Using the same index notation associated with the earlier Laplace system, we shall extend it to label the forcing terms P and Q by P_1 and P_2. A third forcing term will be denoted by P_3 for a third Poisson equation in three dimensions. With an implied summation of repeated indices over their respective ranges, the Poisson system becomes

$$D\vec{r} = 0 \tag{66}$$

where

$$D = g^{ij}\frac{\partial^2}{\partial\xi_i\partial\xi_j} + P_k\frac{\partial}{\partial\xi_k} \tag{67}$$

and $\vec{r}$ is the position vector in the physical space. In comparison to the earlier resultant from the Laplace system in Eq. 60, the forcing terms appear as the second summed component in the operator of Eq. 66. That is the only change. The form for two-dimensional surfaces is given by Eq. 62 with the operator from Eq. 67.

Parabolic Systems

Parabolic systems primarily have arisen as approximations of elliptic systems. While attempting to retain the basic attributes of elliptically generated grids, these methods offer computational speed. As in the earlier hyperbolic methods, parabolic methods are executed by spatially marching away from a body to generate a grid in one sweep. The single sweep provides the basic speed of the hyperbolic methods which in turn is substantially greater than that of elliptic methods. In distinction from hyperbolic methods, the outer boundary can be specified. In comparison with elliptic methods, the controls are generally weaker.

Parabolic methods have been developed by Nakamura (155,156), Edwards (56) and by Hodge, Leone, and McCarty (120).

INTERACTIVE GRID GENERATION

Although a large and diverse number of grid generation algorithms have been developed to generate curvilinear and unstructured grids in two and three dimensions, the time required to produce the desired node distribution over physical space can range from several man minutes to several man months. Traditionally, existing grid generators are applied on a case by case basis. The whole process can be divided into several steps, which are iterated upon. After study of the geometrical configuration to be gridded, a suitable numerical algorithm is chosen. For example, a standard two-dimensional airfoil is easily gridded with a conformal mapping, perhaps followed by a shearing transformation to adjust the nodes along one direction. On the other hand, a standard approach to gridding the space surrounding a fighter aircraft described by a very large number of nodes is to use a multi-block algorithm. Among the questions the user must answer are: whether grid lines should be continuous across the blocks, how many blocks are necessary, how many nodes to place inside each block, how to parametrize the surface boundaries, and what node distribution to use normal to the surface.

Most methods require a subdivision of computational space into regular cells. The desired distribution of nodes on the body surface is achieved either with source functions which become the right hand side of partial differential equations, or with positive weight functions which in one dimension typically satisfy the relation

$$d\xi = w(s)ds \tag{68}$$

along the curve on which nodes are being redistributed. The weight function $w(s)$ is a function of the arclength along the curve. More details are provided in the section on adaptive methods and monitor surfaces. The important point is that without the help of a procedure that automatically chooses the right parameters, the user is burdened with this task. His only recourse is to make decisions based on experience, enter them into the appropriate grid generator package, and submit the program to a mainframe, typically in batch mode. Precious time is lost while the program is recompiled and executed. Eventually, graphic output depicting one or several sections of the grid, with perhaps statistical diagnostics such as average Jacobian deviation from zero and a measure of smoothness is received and analyzed. Based on the results, the input parameters are slightly modified, and the process is repeated.

Two types of codes have emerged to speed up the grid generation process. The first is the integrated package requiring a few selected parameter choices, sometimes with a prescribed density distribution over the physical domain. For example, GRAPE (192), allows the specification of both the node distribution and the spacing along the outward grid line with the surface boundary to be specified in each subdomain. After the first subdomain is gridded, the grid line directions at the outer boundary of the subdomain are imposed on the adjacent domains as initial conditions. This procedure insures continuity of the grid lines across the boundaries of the subdomains. The whole procedure is executed in batch mode. Other examples of automated but batch oriented codes are INMESH for 2-D and EAGLE and NUMESH for the generation of 3-D grids. Further examples can be found, for example in the Landshut Conference (111).

In contrast to batch oriented programs, the proliferation of high-powered graphic workstations with local intelligence strongly suggests an interactive approach to grid generation. Each step of the grid generation is under direct control of the user. The result of parameter changes can be viewed instantaneously on the screen, allowing the user to take immediate action if necessary. Parametric studies can be accomplished at very little expense, and with little risk of error. Special graphic hardware to translate and rotate screen objects in real time is often an integral part of the system. Together with color, the user can examine the grid from all vantage points and quickly detect anomalies. In the span of several hours, grids can be generated that would otherwise have taken days or weeks. Moreover the resulting grids are often of higher quality. Frequent changes to the geometry also make the interactive approach desirable because of the reduced amount of human time since much of the time wasted with batch oriented programs is avoided. To summarize, interactive grid generation allows the user to concentrate on the geometry of the problem rather than on the mechanics of processing programs. Engineers, normally shy of computers, find it easier to interact with a menu driven system, particularly one which provides online help and semi-automatic error detection and correction.

The basic ideas behind four interactive grid generator programs(187,182,66,93) are briefly discussed, and the methods are compared. The first two-dimensional interactive grid generator is that of Smith(187), called the two-boundary grid generation technique. The heart of the algorithm is a transfinite interpolation procedure that combines cubic Hermite interpolation in one direction with a shearing transformation in the other to obtain an exact mapping of the computational domain onto the physical region. Generation of the grid is divided into four steps. First the physical boundaries are read in from a data file, and the number of points on each boundary is specified. The second step involves the parametrization of the physical boundaries. Several specified functions are available to the user, who must supply parameters such as stretching of the coordinate transformation. Should more flexibility be required, the option is available to generate a new mapping. To this end, a graphical representation of the physical domain is drawn along with a graph of ξ versus arclength, s. The user then picks several points on the physical boundary with a cursor (from which s can be determined), with corresponding points along the computational coordinate axis, ξ. A least squares spline is fit to this data and defines the sought after boundary parametrization. Points can be added or deleted in random order. In this way, an optimal boundary node distribution is achieved on all four boundaries. The third step establishes the boundary distribution of derivative quantities required by the Hermite interpolator. As mentioned previously, if the magnitude of these derivatives is too large, the possibility of coordinate curve overlap exists. Only an interactive approach can immediately bring this effect to the user's attention. At this stage the only missing component is the distribution of nodes along the Hermite polynomial. This is done similarly to step 2. Intermediate results can be saved and later restored, letting the grid generation span several sessions. The major disadvantage is the lack of control provided by the Hermite polynomials.

Based on a set of routines suited to the automatic generation of two-dimensional algebraic grids(66), an interactive grid generator, IMAGE was developed at NASA Langley by Erlebacher. The acronym IMAGE stands for Interactive Multisurface Transformation for Arbitrary Grids in Engineering. The added flexibility provided by the intermediate surfaces more than offsets the extra effort the user must undergo to generate the grid when compared to a similar task with the two boundary technique. IMAGE is best suited for complicated O-type meshes, although simple C-mesh structures are also possible. Its great strength lies in the control the user has in the construction of the intermediate surfaces. Superellipses of arbitrary degree, 4 digit NACA airfoils, and polygons entered with a cursor are all available at the touch of a couple of keystrokes. Intermediate surfaces can be created a fixed distance D away from an existing surface S along its inner or outer normal vectors. Not only the boundaries, but all the constructive surfaces must be parametrized. The user has control over the intensity of the curvature pull the nodes will respond to. The higher the intensity, the more nodes get pulled into regions of high curvature (71). Furthermore, intervals along the surface are picked with a cursor, and when used along with the percentage of nodes in the interval, a local clustering capability is provided. Based on the parametrization of the physical surfaces, several routines simplify the parametrization of the intermediate surfaces. Among them, the most useful are the projection operators which generate the distribution of the computational coordinate of one surface based on that of another. Typically, projection methods are only useful between two surfaces close to one another. Otherwise there is a possibility of grid line overlap in the final mesh. The projection operator is often used between the physical body and the first intermediate surface outward to guaranty orthogonality of the coordinate curve with the boundary.

The large number of options available to the user are apt to confuse a novice. Therefore, an online help is available during the grid generation session. Moreover, user errors are reported along with the section of documentation which is pertinent.

Two interactive grid generators, both based on the block structured approach, have been written to treat three-dimensional geometries. Gaitonde (93) has written an interactive program based on the multisurface transformation. It is specialized to a three-dimensional configuration which can be represented as a series of two-dimensional cross-sections. Many aircraft satisfy this constraint. The grid generation problem has therefore been reduced to two dimensions plus the constraint that the coordinate curves normal to the cross sections must not be too skewed. Cross sections with complicated topologies are decomposed into subdomains. The program is menu driven, and provides the user with extensive control over intermediate surface representation and node distribution along all directions.

Finally, a three-dimensional interactive block-structured approach was recently implemented by Seibert (182). Geometry definition, the first task of any interactive program, is accomplished with several available Computer-Aided-Design programs. Libraries of ready made geometrical shapes can be bypassed with the help of digitizers and drawing boards. An added complexity relative to the previous three algorithms is the need to keep track of the relative spatial position of the blocks. Grids are generated either by redistribution of nodes along spline curves, or with a Poisson solver. With this approach, a grid about two intersecting pipes was generated. It is nearly orthogonal almost everywhere. A further application was to grid a fuselage with inlets (182). Again, Seibert managed to generate a nearly orthogonal grid.

For two dimension grids, a little mentioned problem is the interactive construction of boundary curves. Smith (187) developed an elegant approach involving global splines with tension. The tension parameter is user controlled, while the spline is calculated as a best fit to a set of user entered coordinate information. The entire procedure occurs interactively at a terminal, providing the user with instantaneous feedback on the effects of his last modification. Unless the tension parameter is chosen with care, oscillation can occur. While this is not much of a problem for the two-boundary technique, the interactive implementation of the multisurface transformation algorithm involves many more curves to be parametrized. Therefore, a more robust procedure is required. An alternative to standard splines is the local Hermite splines which require a specification of node positions and tangent vectors at these nodes. However, if the tangent vectors are not chosen with care, the interpolated curve can backtrack upon itself and become multi-valued, e.g.

$$\vec{r}(\xi_1) = \vec{r}(\xi_2) \qquad \xi_1 \neq \xi_2 \tag{69}$$

This phenomenon is observed by Smith (187) in the two-boundary technique.

A major hurdle that all interactive grid generators must overcome is the automatic generation of oscillation-free interpolating boundary surfaces. Various methods have been proposed. A simple technique, not proven to be oscillation-free, but observed to be so in practice, uses the property that along the interpolated curve, the magnitude of the derivative of the vector $\vec{r}$ with respect to arclength s is always unity. Therefore, if ξ_n is the cumulative linear arclength between $\vec{r}_0$ and $\vec{r}_n$, a natural definition of the tangent vector between $\vec{r}_n$ and $\vec{r}_{n+1}$ is

$$\vec{\tau}_n = \frac{\vec{r}_{n+1} - \vec{r}_n}{\xi_{n+1} - \xi_n} \tag{70}$$

of unit magnitude. In terms of $\vec{\tau}_n$, the tangent vector $\vec{t}_n$ at the node $\vec{r}_n$ is defined by

$$\vec{t}_n = \frac{1}{2}(\vec{\tau}_{n-1} + \vec{\tau}_n) \tag{71}$$

whose direction coincides with one of the bissectors of the angle θ formed by the vectors $\vec{r}_n - \vec{r}_{n-1}$ and $\vec{r}_{n+1} - \vec{r}_n$. The magnitude of $\vec{t}_n$ is $\cos(\theta/2)$. Being less than unity (which is the analytical value), it is highly unlikely that oscillations will occur. When θ is very close to π, the tangent vector magnitude approaches zero which is a highly desirable feature. Standard splines will produce severe oscillations about such a sharp corner. With local Hermite polynomials, setting the tangent vector to zero at sharp corners heavily dampens the oscillations that would otherwise occur. Many tests of this approach were performed on severe boundaries, including fuselage cross-sections of advanced aircraft configurations. They all produced boundary fits with no detectable oscillations.

ADAPTIVE MESHES

In a variety of important physical phenomena, there is some distinctive or critical attribute which is changing rapidly at some unpredictable location in space and also possibly in time as well. With fixed meshes, such phenomena are often not adequately captured in numerical simulations, at least without an inordinate number of mesh points. Even a large number of points may still not be satisfactory since the severity of the rapid variations may unpredictably become excessive. To rationally consider accurate solutions in these circumstances, adaptive mesh techniques have evolved. In broad terms, the techniques consist of either adjusting the number of points or moving them. On some occasions, there is some mix of techniques such as movement with either local or global changes in the number of points. With structured grids, the most reasonable blend is the global one whereby the total number of points in a coordinate direction can be adaptively altered. With unstructured meshes, by contrast, the local one is preferred.

The essential ingredients in adaptive mesh methods are a means to adequately monitor the severe solution behavior and a means to appropriately utilize that data. The monitoring aspect is formally consolidated into a single simply defined object: to be descriptive it is called a monitor surface. The utilization of that surface data consists of a mesh generation part, a PDE-solver part, and a coupling part. While the various parts can be assembled in a variety of ways and at distinctive times or sequences, we shall concentrate on the mesh generation part and give indications of how the other parts relate to it. The mesh generation part is further split into structured and unstructured formats.

For a more thorough and in depth discussion of adaptive grid generation, the reader is referred to the review by Eiseman (57). Earlier reviews are also available and are given by Thompson (226) and Anderson (4). Moreover, adaptive grids have been considered in a number of conferences (223,97,111,11), general review papers (221,224,69) and a text (225).

MONITOR SURFACE

In a formal sense, at any point in time, the best numerical grid minimizes the truncation error of the discretized differential operator to be inverted. However, even when feasible, the determination of truncation error formulae sufficiently accurate to be of use is unwieldy at best. The formulas obtained are very much dependent on the numerical scheme adopted and the upper bounds on the errors might not be tight enough. Furthermore, when systems of equations are involved, one is left with a collection of truncation error estimates, and the problem of which combination to minimize must still be tackled.

An alternative view is to pinpoint one or several variables which drive the physics of the system of interest, and resolve them instead of the error. The reasoning is very simple. If the physical variable is resolved properly, the truncation error of the differential equation must be low, otherwise, the physical variable would be inaccurate. The strategy is therefore to insure that at the initial time, the grid is sufficiently fine to accurately represent those variables. As time evolves, the grid is adapted to the smooth, time-dependent changes in the variables, and therefore accuracy is maintained. For example, in shock dominated flows, pressure is a representative variable because it is very sensitive to shock locations, which are the bane of many numerical methods. If the initial flow is shock-free, it is relatively simple to track the spatial variations in the pressure field, and to add grid points, or cluster grid lines in areas where the gradient or curvature of pressure are above specified thresholds. This approach is independent of the numerical algorithm adopted to solve the problem since the grid movement is typically decoupled from the solution step of the physical equations.

In general, there may be more than one variable one wishes to resolve. A typical example is the multiple species in a combustion process (50) where a number of variables must be tracked if their effect on the flow is to be determined accurately. Let the number of salient quantities the grid must adapt to be denoted by N. These variables form an N dimensional vector, whose components are in turn functions of the n physical coordinates of the grid points. Geometrically, this vector represents a surface in an imbedding space of dimension $N + n$. One possible approach to constructing a suitably adapted grid, is to somehow generate a grid on this surface and project down onto the physical space. The standard approaches to node attraction, namely gradient and curvature pull, are hard to apply in spaces where N is greater than one. This is because variations in the surface normal direction, typically used in curvature evaluation, is not a uniquely defined quantity. For example, when $N = 2$ and $n = 1$, the two salient variables can be written as

$$V_i = V_i(x) \qquad (i = 1, 2), \tag{72}$$

which can be interpreted as a curve imbedded in the three-dimensional space (x, V_1, V_2). As is well known, curvature is not sufficient to describe this curve. The added required parameter is the curve torsion. However, instead of independently tracking the variations of V_1 and V_2 as a function of x, it is often useful to linearly combine them into the single function

$$V = \alpha_1 V_1 + \alpha_2 V_2 \tag{73}$$

where α_1 and α_2 are user defined parameters. As long as the variations in the two variables V_1, V_2 do not cancel each other out, it is possible to resolve both variables by attracting nodes according to the geometry of (x, V). This entity is simply a curve in a two-dimensional embedding space, geometrically characterized by its curvature (69). It is clear that it is far simpler to attract nodes based on the geometry of x, V than that of x, V_1, V_2 because of the reduced amount of computation required.

The process of combining the N quantities into a single scalar combination becomes increasingly attractive for large numbers of quantities. Whereas the analytical formulas involved in the grid generation also become more tedious as the dimension of the physical space increases, there is no corresponding simplifying mechanism. The resulting scalar quantity, which the grid points must adequately resolve is called a monitor surface.

For one-dimensional problems, the monitor surface becomes a curve. Two points of view prevail. The monitor surface can either be gridded directly, and the grid projected onto the coordinate axis, or geometrical information from the surface can be used to generate a grid on the coordinate axis. The first approach has the advantage of not requiring derivative information since it is built into the surface. In the absence of high curvature, a uniform surface grid projected onto the coordinate axis naturally clusters points in regions of high gradients. When the grid is generated on the coordinate axis, however, the clustering formulas must explicitly include surface gradient magnitudes to accomplish a similar clustering. Curvature and additional clustering parameters can added to the surface grid generation algorithm to redistribute surface points in accordance with the adaptivity requirements of the physics.

When the monitor surface exhibits severe bends, surface curvature must be taken into account. For one dimensional problems, the curvature is given by the simple formula

$$\frac{V_{xx}}{(1 + V_x^2)^{3/2}}. \tag{74}$$

However, a common approximation made for curvature is to set it equal to a second derivative. This approximation is only valid when the norm of the gradient vector is much less than one. If the gradient norm and the absolute value of the second derivative are high, the curvature decreases because of the gradient norm in the denominator. Accordingly, the exact formula given by Eq. 74 must then be employed.

In two and higher dimensions, the situation becomes more complex. It is no longer obvious what constitutes a good grid in the monitor surface (or volume) since there are more than one coordinate direction. Equidistribution of cell volumes, aspect ratio or conformality are all valid criteria for constructing the surface grid. A more thorough discussion of monitor surfaces can be found in Eiseman (69). An application of monitor surfaces in plasma physics on unstructured meshes using two salient quantities has been reported in Erlebacher (79,80).

ADAPTIVE GRIDS

Adaptive grids are necessary when the significant solution variations are to be resolved in a structured manner. The structure comes from coordinate transformations and is reflected in the connectivity pattern as discussed earlier. The consequent regular ordering of points is important because the numerical algorithms for solving partial differential equations are generality more efficient and can often be optimized. With the maintenance of the connectivity structure, the primary adaptive action is grid point motion. The inherent coordinate transformations are then evolutionary in character and provide a distribution of points that conform very closely to the significant solution variations. When the distribution is appropriately determined, further adjustments then come from altering the total number of points in the various coordinate directions. This keeps the same transformations, but adjusts the number of points to optimize between under and over resolution from a global perspective. In our discussion, the central aspect of determining the distribution will be explored. Accordingly, various grid point movement schemes will be described from the viewpoint of using only a fixed number of points.

With the assumption of a monitor surface, we may proceed to consider grid generation methods that are sufficiently reliable and automatic to provide the desired motion while retaining a decent grid structure. These methods can be developed in either of two natural ways: either the basic grid generation is performed on the surface or it is done in physical space. If the generation occurs directly on the surface, then the physical space grid is just the downward projection of the surface grid. The advantage of the surface grid generation is that a uniform surface grid automatically resolves the gradients in physical space. As a consequence, the demands upon weight functions are light. In contrast, the basic grid generation in physical space is simpler but the demands upon weight functions are heavier.

One Dimension

Regardless of the location where the grid generation is performed, the constructive principles remain the same. Moreover, virtually all of the basic methods [illegible]ts in the one-dimensional setting. In that context, we can then understand the basic elements without undue mathematical detail. As a consequence, we are drawn to consider the process in which a weight function is to be equally distributed between the successive points of a grid on a curve. If two successive points are denoted by their arc length positions s_i and s_{i+1}, and if the averaged weight function between them is denoted by $w_{i+1/2}$, then the equal distribution is obtained when

$$w_{i+1/2}(s_{i+1} - s_i) = cst \tag{75}$$

is satisfied for all i. The desired effect, here, is that large values of $w_{i+1/2}$ cause the spacing between s_i and s_{i+1} to shrink. That is, points are attracted to regions of larger weight. In terms of coordinate transformations, this can be restated in the differential form

$$wds = cd\xi \tag{76}$$

where c is a constant and ξ is the curvilinear variable that is given constant increments. The constant c can be removed by differentiation to yield the differential equation

$$s_{\xi\xi} + \frac{w_\xi}{w} s_\xi = 0. \tag{77}$$

In effect, the constant is shifted from the global statement into an additional boundary condition. The transformation can be explicitly stated from a direct integration of Eq. 76 or implicitly stated in tridiagonal form from Eqs. 75 and 77. Two successive values of i are used for Eq. 75 while an immediate discrete approximation is employed for Eq. 77. With the tridiagonal forms, point relaxation in a mean value sense is readily seen. Moreover, variational forms that have Eq. 77 as the Euler equation can also be established. Each of the mentioned forms extends into the multidimensional context.

An important aspect of each form is the dependency of the weight function: it can be a function of either the physical position s or the grid point counter ξ. Since the adaptive data appears with respect to only physical position, the use of a weight function dependency upon ξ requires iteration while a dependency upon s does not. This distinction occurs regardless of whether the basic method is already iterative or not. However, if it is, the requirement appears more modestly by including the weight function in the existing updating process. In contrast, for example, a global integration of $1/w(\xi)$ in Eq. 76 directly yields an entire transformation that must be iterated. At each stage, the weights are computed from the current grid point locations s_i and are considered as functions of i which is effectively ξ. The integral transformation then determines new positions s_i and the process is repeated.

Curve By Curve Methods

With a basic understanding established, we next proceed to consider higher dimensions. The most direct extension into higher dimensions is to apply adaptivity on a curve by curve basis and to cycle through one or more coordinate directions. The methods of this description are called "alternating direction adaptive methods." From a geometric viewpoint, these methods operate by specifying the diagonal part of the metric tensor since each such diagonal entry is inversely proportional to a specified weight. This fundamental geometric framework provides a unified setting for all such methods and was presented by Eiseman (67). The unity, here, derives from the fact that no matter how the metric specification is accomplished, the result must be the same up to the distinctive errors of approximation. The completeness in the specification comes from the curvature of space. This is most readily seen in two dimensions where Gaussian curvature provides a single relationship between all three metric coefficients g_{11}, g_{22}, and g_{12}. That means that the coordinates are analytically determined when g_{11} and g_{22} are specified as positive functions that may even involve g_{12}. In other words, the two available degrees of freedom are employed. Likewise, in three-dimensions, the three available degrees of freedom are consumed. The typical specifications include gradient magnitudes, various forms for curvature (normal, geodesic, mean, second derivatives), cell properties (eccentricity, aspect ratio, lengths, area or volume), and the attraction to prescribed distributions (uniform, arbitrary, orthogonal alignment, or previous locations).

While the specifications can be accomplished with a wide variety of weight function formulations, the simplest and most commonly employed form is linear. Each of the specified quantities are represented by non-negative functions that are viewed as mass distributions M_k along the current curve. Assuming an application relative to uniform conditions, we set the first mass M_0 equal to 1 and get the linear form

$$w = 1 + c_1 M_1 + ... + c_m M_m \tag{78}$$

where the nonnegative coefficients c_k give the relative importance of the k^{th} quantity represented by M_k. This weight function can be employed with any of the forms for equidistribution that come directly from Eqs. 75-77.

In the higher dimensional context, Eq. 76 becomes the basic metric statement

$$g_{jj} = (\frac{c_j}{w_j})^2 \tag{79}$$

where the subscript j for the coordinate direction is also attached to both the constant and the weight. For each given curve c_j is a constant as earlier. However, as we go from curve to curve within the family for a given direction, that constant becomes a function of the transverse variables which govern this progression. As a consequence, differentiation of $g_{jj}w_j^2$ in Eq. 79 will dispose of the constant and will yield the partial differential equation

$$\frac{\partial g_{jj}}{\partial \xi_j} + \frac{2}{w_j}\frac{\partial w_j}{\partial \xi_j} g_{jj} = 0 \tag{80}$$

that corresponds to the earlier ordinary differential equation of Eq. 77. As in the case of Eq. 77, the data from the constant has been effectively transferred into boundary conditions. Upon substitution for g_{jj} in Eq. 80, the partial differential equation is readily observed to be second order. To be explicit on this matter, we consider the case of the first equation in two-dimensional Euclidean space. There, the metric from Eq. 3 is just $g_{11} = x_\xi^2 + y_\xi^2$ and results in

$$x_\xi x_{\xi\xi} + y_\xi y_{\xi\xi} + \frac{(w_1)_\xi}{w_1}(x_\xi^2 + y_\xi^2) = 0. \tag{81}$$

The equation for the η direction has the same form and is obtained similarly. This specialized case was presented by Anderson (4). Other forms cover curves on surfaces and higher spatial dimension. All of these come from Eq. 80. In comparison with the earlier case of an isolated curve, the equations, here, are for coordinate curves, and as a consequence, reflect their embedding within the coordinate system.

To examine the historical roots of alternating direction adaptivity, we return to one dimension and then expand to higher dimension and to multiple directions. Some of the earlier studies in one dimension were developed by Winkler (242), Ablow and Schecter (1), and White (240,241). Winkler (242) considered grid point movement directly in physical space in response to gradients in the monitor surface. All attributes for clustering and grid structure were given in the form of a linear weight. By contrast, White (240) developed his grid directly on the solution curve and thereby effectively used a weight of unity. When the arc length was expressed as an explicit function of physical space, he obtained the appropriate weight function for arc length. In extrapolating from here, he called the weights monitor functions. Although, the term may be appropriately descriptive when everything is combined under a single integral, it is otherwise deficient. A more basic consolidation of the data and one that applies regardless of dimensionality is the concept of monitor surface as discussed earlier herein. Ablow and Schecter (1) preceded White and considered a linear weight with curvature that was applied relative to the arc length of the solution curve.

In the next stage of development, Dwyer et al. (50,51,53,54,52) considered the process of adapting the points along each coordinate curve in a fixed direction. In contrast to Winkler, White and Ablow and Schecter, he considered weight functions that depended upon positions in physical space. This was executed in a noniterative fashion by a direct integration of Eq. 76 along each coordinate curve in the physical region. Again, linear weights were employed. In terms of Eq. 78, he considered up to two masses consisting of magnitudes for first and second derivatives of the monitor surface. These, however, provided only approximate gradient and curvature data: the variations along transverse coordinate curves were ignored. An advantage that evolved from the choice of linear spatially dependent weights was the capability to more rationally define the coefficients in the linear weights. With the transformation established by the global integrals, Dwyer (53) noted that the fractional contribution of each mass was merely a ratio of that mass integral to the total mass integral. Each integral, of course, was taken over the entire curve. Since each mass integral contains the associated coefficients, he was able to specify the fractions and solve for the coefficients. This was done for up to two masses. As a consequence, each specified fraction resulted in a weight coefficient that dynamically adjusted from curve to curve.

In progressing from a single direction to multiple directions both Ablow (2) and Gnoffo (101) performed bidirectional studies. Ablow considered the arc length along the curves on a monitor surface and proceeded to solve the equations

$$\begin{aligned} x_\xi x_{\xi\xi} + y_\xi y_{\xi\xi} + z_\xi z_{\xi\xi} &= 0 \\ x_\eta x_{\eta\eta} + y_\eta y_{\eta\eta} + z_\eta z_{\eta\eta} &= 0 \end{aligned} \tag{82}$$

which come from Eq. 80 with constant weights and with $z = z(x, y)$. He employed an ADI procedure for the solution. As a consequence, he automatically clustered to high gradients. In contrast, Gnoffo viewed the monitor surface only from physical space, and thus had to explicitly use gradient information. Like Dwyer (51), he neglected transverse variations and considered derivative magnitudes along the given coordinate curve in physical space. He did not, however, consider derivatives beyond first order. In the execution phase, the weight was viewed as a function of the grid point index (equivalently ξ or η) and formed a global integral that required iteration as discussed earlier. In the trapezoidal quadrature rule for the integral, each increment in ξ and η was unity and the result was a simple sum of reciprocal weights. This is in contrast to the noniterative statement where the quadrature is a sum of products between midpoint weight values and the corresponding interval arc lengths. As a matter of terminology, he called this approach a spring analogy. Because of the iterative nature, however, the *spring constants* are not really constants since they must also change. At best, they may be viewed then as nonlinear springs.

In a more general study, Eiseman (67) consolidated and extended the previous works and presented a mathematical foundation for all such curve by curve methods. To be descriptive, these methods were then called alternating direction adaptive methods. The mathematical foundation was mentioned earlier. In brief terms, the known curvature of a region implies that metric specifications along each coordinate direction are enough to completely determine the metric which in turn can be employed to generate the grid by line integration. In the context of orthogonal grid generation, details on the line integration for coordinate positions can be found in Warsi and Thompson (237) and in Eiseman (65).

With the previous works separated between surface grids without weights and planar grids with weights, the more unified approach is to simultaneously consider surfaces and weights. In the consolidated form, a stated preference was given for generating grids on monitor surfaces while using weights for the resolution of surface properties; albeit, the form applies equally well to viewing the surface from the physical region below it. The stated preference quite naturally derives from the fact that the accurate representation of a surface is more readily apparent in the surface grid than in the corresponding projected grid in the physical region. The reason is that the viewpoint is the location of arc length measurement which is more naturally taken and accurately controlled on the surface. In simple terms, it is reasonable to generate the grid directly upon the very object that must be given a good representation.

With the objective of providing a good representation for the monitor surface, the geometric parameters for the surface must be used along with the parameters which govern the grid quality in the sense of good structure. The basic surface properties are the bends or folds in the surface and are the surface boundaries which may also be bent in some manner. The basic measurement for surface bending in a given direction is the normal curvature. The measurement for the boundaries is the geodesic curvature. Each curvature detects only the desired property. Using the normal curvature in the direction of the current coordinate curve, the desired clustering effect for surface folds was observed. In addition, the use of normal curvature was seen to provide some alignment of coordinate curves with folds in the surface. Moreover, the formulation used the general linear weight of Eq. 78 where normal and geodesic curvatures are balanced with the unity for uniformity and a mass for orthogonality attraction. The coefficient for geodesic curvature was presented in a form that decayed upon leaving boundaries so that bent interior curves appearing from the iteration would not unnaturally cause clustering.

The use of fractional specifications initiated by Dwyer (53) were also extended. In a basic sense, it was noted that the resolution of a property which appears over a fixed length could be depreciated by merely increasing the total length (or other masses). Thus a mechanism was required to appropriately adjust the clustering intensity to treat the local property in the same manner, regardless of length. For normal curvatures we must somehow detect the likely presence of local clustering regions and the possible excessive consumption of surface arc length. This was done by first forming the ratio R of the actual arc length of a coordinate curve on the surface to the minimum possible length. The latter is computed in the form of a Euclidean distance using the arc length of the projected curve and the change in altitude between endpoints. Upon application, a factor of the form $\tanh[D(R-1)]$ was applied to a constant fraction determined in the original manner. The fraction is now seen as a maximum possible fraction that would be employed in the most severe case within the family of all coordinate curves in a given direction. The constant D then gives the rapidity for which the specified maximum fraction is approached. More details are provided in Eiseman (67). Another extension of fractional specifications is to include any number of masses in Eq. 78 and to consider what happens when distinct masses appear on the same interval. This is discussed in the reviews by Eiseman (69,57) and is not repeated here.

In three subsequent studies, Nakahashi and Deiwert (151,154,152) added a few more items of interest to the development of alternating direction adaptivity, and in addition, presented some rather good examples of adaptive simulation in aeronautics. The main contributed item is their incorporation of an orthogonality control outside of the weights. Given the arc length locations r_i corresponding to an orthogonal alignment with the previous curve, they added a term of the form $D(s_i - r_i)$ to the right-hand side of the tridiagonal system derived from Eq. 75. Altogether, the new tridiagonal form is given by

$$w_{i+1/2}s_{i+1} - (w_{i+1/2} + w_{i-1/2} + D)s_i + w_{i-1/2}s_{i-1} = -Dr_i \qquad (83)$$

As D increases, the diagonal dominance grows and forces s_i to approach r_i. A variant is to make D change with i (D_i). In the applications, D_i actually varied inversely with respect to the distance from the previous curve at each i. This strengthened the attraction for closely spaced curves.

In three dimensions, there is another similar arc length location t_i and another term $E_i(s_i - t_i)$. There, $D_i + E_i$ is added to the diagonal and $D_i r_i + E_i t_i$ forms the right hand side. In continuing the analogy with springs, they attributed the orthogonality control to torsion springs rather than to diagonal dominance. Although this control might at first appear to make the method distinctive, the fundamental fact still remains that a metric relationship is being specified along curves. In fact, if the location r_i in Eq. 83 is given by the relative location

$$r_i = \alpha_i s_{i+1/2} + (1-\alpha_i) s_{i-1/2} \tag{84}$$

through α_i, then we get new adjusted weights

$$\begin{aligned} \overline{\omega}_{i+1/2} &= \omega_{i+1/2} + \alpha_i D_i \\ \overline{\omega}_{i-1/2} &= \omega_{i-1/2} + (1-\alpha_i) D_i \end{aligned} \tag{85}$$

where the second term can be interpreted as a mass for orthogonality attraction as discussed in regard to the general linear weight of Eq. 78. If r_i lies between $s_{i-1/2}$ and $s_{i+1/2}$, then α_i is between 0 and 1.

Rather than a consistent use of linear weights, Nakahashi and Deiwert (152) considered weights where some of the specified constants appeared nonlinearly. This arose mainly because the minimum and maximum spacing could be specified by means of a single algebraic formula over the curve. One constant was an exponent and had to be determined by iteration. By contrast, Winkler (244) gave upper and lower bounds upon the spacing within the context of the general linear weight of Eq. 78 and did not require an iterative determination of constants. This was possible because the spacing was only required to become close to minimum and maximum spacing along the curve rather than matching it precisely.

It should be noted that the objective in setting bounds upon the spacing as represented by Winkler (244,243) and by Nakahashi and Deiwert (152) is essentially the same objective as specifying the fractional amount of quantities as represented by Dwyer (53) and Eiseman (67). Both prescriptions merely attempt to set limits upon the finite distribution of points. In comparison, the spacing bounds are attractive because of their direct attachment to the actual spacing while the specification of the fractions are attractive because of their flexibility. With the fractions, the constraints upon spacing can be more effectively balanced against the other attributes which quite naturally enter into the same linear format. Moreover, those other attributes can also be precisely separated at the same time as the spacing requirements. This extensibility is not readily apparent in the format of spacing bounds. To consider it, the form presented by Winkler (244) would be preferred.

In the studies of Nakahashi and Deiwert (151,154,152) no defect was noted in the application of the method. The only indication appears indirectly when they state that some corrective action is required when the orthogonally aligned arc length locations r_i or t_i fail to be monotone. Although the action was simply executed, the real problem was covered up. The real problem comes from the errors incurred in the numerous piecewise linear approximations to curves and their parameterizations. Various forms of the problem were illustrated by Eiseman (67) along with certain explicit corrective actions. In cases with rapid but not excessive variations, such actions, however, are usually not required.

In summary, while the straight curve by curve adaptation yields good results in many circumstances, there is a significant underlying limitation on the weights. Namely, the weights cannot be too severe or else the procedure will collapse. This has been observed and while corrective action can be inserted directly into the process, such action is rather detailed and technical.

In a further study, Eiseman (72) found that a better course of action is to redefine the directional sweeps by splitting them into two phases: the active phase and the passive phase. In the active phase we just have the original curve by curve strategy in the current direction. This contains the fundamental adaptive forces. In the passive phase, a *low pass filter* is applied to remove any wiggles or abrupt changes in spacing caused by the active phase but to leave intact the basic results of the intended action. This produces a smooth grid in the sense of derivative continuity. As a consequence, continuous numerical derivatives are available for numerical solution algorithms and for the application of controls in successive sweeps. Such controls include the use of orthogonality and curvature in the weights. As a practical matter, it has been observed that the splitting of sweeps into this predictor-corrector format of active and passive phases has resulted in considerably enhanced stability and a much larger range of severity in the choice of weights.

The simplest form for the passive phase is given by the direct action of a Laplace filter upon the grid point locations. While such action by itself may not be appropriate for the generation of a nonsingular grid, it is certainly suitable for the stated purpose of establishing derivative continuity. The distinction is that the filter is applied at most a few times in each directional sweep rather than being driven towards convergence as is the case when the solution to a system of grid generation equations is sought. The Laplace filter is given by the simple Gauss-Seidel relaxation of

$$\begin{aligned}\vec{r}r^{n+1}_{i,j,k} = \vec{r}r^{n}_{i,j,k} \quad + \quad & \frac{1}{12}[\vec{r}r^{n}_{i+1,j,k} + \vec{r}r^{n+1}_{i-1,j,k} + \vec{r}r^{n}_{i,j+1,k} \\ & + \vec{r}r^{n+1}_{i,j-1,k} + \vec{r}r^{n}_{i,j,k+1} + \vec{r}r^{n+1}_{i,j,k-1} - 6\vec{r}r^{n}_{i,j,k}]. \end{aligned} \tag{86}$$

At the boundaries, this formula was restricted to provide filtering in only tangential directions. In the application, a three-dimensional monitor surface was defined in four-dimensional Euclidean space by means of the vector (x, y, z, u) where u is considered to be a function of (x, y, z). To obtain the physical space projection, we merely replace u with a constant which is usually 0. In a test of the basic movement, u was taken to have a severe variation across two intersecting ellipsoids that also intersected the boundaries of a Cartesian box. From an initial surface grid with a Cartesian grid projection, an equal arc length grid was rapidly generated on the surface, and accordingly, a smooth gradient clustered grid resulted in the (x, y, z)-projection $(x, y, z, 0)$.

While the passive phase of each sweep provided smoothness in the sense of derivative continuity, another measure of smoothness is provided by an attraction to conformal conditions. This latter measure is more demanding and is thus clearly stronger. Recognizing the basic need for smoothness, Anderson and Steinbrenner (5,6) brought the process of equidistributing weights along coordinate curves into the format of the Poisson system of Eq. 63. Their development was motivated by the previous work of Middlecoff and

Thomas (143) who employed the formulation

$$g_{22}(\vec{r}r_{\xi\xi} + \phi\vec{r}r_{\xi}) - 2g_{12}\vec{r}r_{\xi\eta} + g_{11}(\vec{r}r_{\eta\eta} + \Psi\vec{r}r_{\eta}) = 0 \tag{87}$$

with

$$\begin{aligned} P &= \frac{g_{22}}{g}\phi \\ Q &= \frac{g_{22}}{g}\Psi \end{aligned} \tag{88}$$

for the purpose of converting boundary distributions into forcing functions thereon. While Middlecoff and Thomas interpolated the forcing functions into the field so that boundary distributions could be propagated into the interior of the field, Anderson and Steinbrenner viewed each curve as if it were such a boundary curve. In the boundary application, the assumption of orthogonality and vanishing transverse curvature was employed to get the differential equation for an equidistributed weight. In particular, with $g_{12} = \vec{r}r_{\xi} \cdot \vec{r}r_{\eta} = 0$ and $\vec{r}r_{\eta\eta} = 0$, the equidistribution statement of Eq. 77 is obtained from Eq. 87 as

$$s_{\xi\xi} + \phi s_{\xi} = 0 \tag{89}$$

where s is the curve arc length and

$$\phi = \frac{w_{\xi}}{w} \tag{90}$$

is the relationship between the forcing function ϕ and the weight function w. In the more general interior application, the same sort of equidistribution statement was established without the previous assumptions. The main distinction is the addition to ϕ of a term for orthogonality and a term for curvature. These orthogonality and curvature terms represent the attraction to the conformal measure of smoothness. Without forces, they yield the desired smoothness. With forces, a deviation is obtained.

In the adaptive context, when the local force is sufficiently strong, the equidistribution force overpowers the smoothness conditions to become dominant and, thereby, to provide the desired equidistribution of the weight along each curve. The equidistribution is more localized than the previous derivative continuous measure of smoothness. This occurs because the equidistribution is essentially cut off unless it is sufficiently strong. The value of such a cutoff is that intense local clusters can be formed primarily from curves that are not too far from the given locality. In contrast, the actual equidistribution of weights along curves adjusts all points along curves, and thus, tends to globally propagate adjustments to intense local requirements. This tendency can of course be limited with the explicit use of orthogonality and curvature attraction. A similar development without an explicit injection of equidistribution was also given by Hāuser, Papp, Eppel, and Sengupta (110).

Finite Volume Methods

Finite volume methods are methods where the grid point motion is based upon the volume elements between grid points. In the context of the finite equidistribution statement of Eq. 75, the direct higher dimensional extension is given by a center of mass formula. For simplicity, only two-dimensions will be considered since the fundamental pattern is established therein. A two-dimensional stencil centered about a grid point $\vec{r}_{ij}$ is assumed to contain only the nearest neighboring grid points. Altogether, four quadrants are determined about $\vec{r}_{ij}$. In standard succession, the first quadrant is subtended by the $i+1$ and $j+1$ direction; the second, the $i-1$ and $j-1$ directions; the third, the $i-1$ and $j+1$ directions; and the fourth, the $i+1$ and $j+1$ directions. For each quadrant, the associated volume is taken to be the triangle attached to $\vec{r}_{ij}$. For example, the first quadrant volume is delineated by $\vec{r}_{ij}$, $\vec{r}_{i+1,j}$, and $\vec{r}_{i,j+1}$. The local motion of $\vec{r}_{ij}$ comes from the weighted volume elements which are defined by the triangles containing $\vec{r}_{ij}$. In each quadrant k, a barycenter $\vec{b}_k$ and a weighting quantity, q_k is obtained as the average of positions and values respectively over the triangle vertices. With the quantity q_k uniformly distributed over the triangle area v_k, the weight w_k at the barycenter $\vec{b}_k$ is just $q_k v_k$. The most common form for q_k is simply the general linear form given by Eq. 78. From the triangles with weights w_k centered at $\vec{b}_k$, the center of mass of all four quadrants is given by

$$\vec{r}_{ij}^{\,new} = \frac{\sum_{k=1}^{4} w_k \vec{b}_k}{\sum_{k=1}^{4} w_k} \tag{91}$$

to determine the new position for $\vec{r}_{ij}$. The actual movement is given by the vector difference $\vec{r}_{ij}^{\,new} - \vec{r}_{ij}$ and is employed in a point iterative cycle. This cycle can be reasonably called *mean value relaxation.*

Like the earlier Laplace system, the motion represents an attraction to conformal conditions when each w_k is the volume v_k itself. In terms of the linear weight form of Eq. 78, the first term which is the number 1 is then the representative of the conformal attraction relative to which the other forces are applied. The analytical indicator for the conformal conditions comes from the converse of the mean value theorem which is simply presented by Epstein (75) on pages 146-148. While the analytical argument employs the area mean value of functions over circular disks, the finite parallel given by Eq. 91 with $w_k = v_k$ is only an approximate form employed within an iterative cycle.

The pointwise relaxation of the center of mass formula of Eq. 91 has been considered by Diaz, Kikuchi and Taylor (48), Oden, Devloo, and Strouboulis (158), Schwartz and Connett (181), Connett, Agarwal, and Schwartz (40), Erlebacher and Eiseman (80), Eiseman (70), and Erlebacher (79). In the studies by Erlebacher and Eiseman, the more general application to unstructured meshes was developed. More discussion is given in the section on adaptive triangular meshes.

In the center of mass formula of Eq. 91, the control over the grid comes from the specification of a single weighting quantity that is applied against the triangular elements of area. While this approach yields a control over the elemental area distribution, it does not exercise all of the available degrees of freedom for such control. Yet one more degree of freedom is available. The full utilization of all degrees of freedom was evident in the transition from the Laplace system of Eq. 54 to the Poisson system of Eq. 63. A similar transition is a reasonable expectation for the process of mean value relaxation.

The utilization of all degrees of freedom in mean value relaxation was developed by Eiseman (68). As in the Poisson system of Eq. 63 and in the alternating direction methods of adaptivity, control was established with a coordinate directional bias that provided the required separation into distinct degrees of freedom. In the local stencil of figure 1, the directional bias was obtained by a projection of the movement vector onto the coordinate curve through $\vec{r}_{ij}$ in the appropriate direction. If w is the weight for the i-direction, then the projection is onto the curve from $\vec{r}_{i-1,j}$ to $\vec{r}_{i+1,j}$. In the implementation, the transverse coordinate curve from $\vec{r}_{i,j-1}$ to $\vec{r}_{i,j+1}$ was used to divide the weights so that first and fourth quadrants would pull towards $\vec{r}_{i+1,j}$ from $\vec{r}_{ij}$ while second and third quadrants would pull towards $\vec{r}_{i-1,j}$ from $\vec{r}_{ij}$. On each side of the transverse coordinate curve, a center of mass was computed and then projected onto the appropriate segment from $\vec{r}_{ij}$. Denoting the projected distances by d_+ and d_- for the positive and negative i-directions from $\vec{r}_{ij}$, the new position along the curve is given by

$$d = \frac{w_+ d_+ + w_-(-d_-)}{w_+ + w_-} \tag{92}$$

where

$$\begin{aligned} w_+ &= w_1 + w_4 \\ w_- &= w_2 + w_3. \end{aligned}$$

Since the construction is centered at $\vec{r}_{ij}$, the sign of d gives the direction of movement: negative is towards $\vec{r}_{i-1,j}$, while positive is towards $\vec{r}_{i+1,j}$. In a similar manner, a second weight $\overline{w}$ is employed for the j-direction and a similar distance is determined along that direction. The signs of the two distances then determine the quadrant which contains the new position. That new position is determined by interpolation.

In the original form, Eiseman (68) considered a bilinear interpolation that also included the diametrically opposite point to $\vec{r}_{ij}$. In a further study, Schwartz and Connett (181) and Connett, Agarwal, and Schwartz (40) experimentally found that the method became unstable when the weights became sufficiently severe. While this led them to consider the earlier center of mass form of Eq. 91, it led Eiseman to consider barycentric interpolation in place of the original bilinear interpolation. The intuitive reason was that the asymmetric use of diametrically opposite points would be more restrictive on the stability. In subsequent tests, the intuition was confirmed: the use of barycentric interpolation led to stable results even when the weights had considerable severity.

Rather than a direct construction of a finite difference formula as in mean-value relaxation, the local volume elements can also be used to form a sum of squares that is minimized to produce a grid. Although this is a variational format, Kennon and Dulikravich (129,128), and Carcaillet, Kennon and Dulikravich (33,34) pursued this topic by using an optimization technique in place of an Euler equation approach. The primary motivation for this minimization problem came from previous variational studies (30,29,31,176) along with the desire to place a larger constant at the center location $\vec{r}_{ij}$ than would have come from the traditional use of central differences that would have missed $\vec{r}_{ij}$.

The basic attractive forces come from distinct sums of squares that are balanced against each other. Of the forces, the first one pulls towards an equal volume distribution with terms of the form

$$(v_1 - v_2)^2 + (v_2 - v_3)^2 + (v_3 - v_4)^2 + (v_4 - v_1)^2 \tag{93}$$

for each point $\vec{r}_{ij}$. The second one pulls towards an orthogonal grid with terms of the form

$$(\vec{A} \cdot \vec{B})^2 + (\vec{B} \cdot \vec{C})^2 + (\vec{C} \cdot \vec{D})^2 + (\vec{D} \cdot \vec{A})^2 \tag{94}$$

where

$$\begin{aligned} \vec{A} &= \vec{r}_{i+1,j} - \vec{r}_{ij} \\ \vec{B} &= \vec{r}_{i,j+1} - \vec{r}_{ij} \\ \vec{C} &= \vec{r}_{i-1,j} - \vec{r}_{ij} \\ \vec{D} &= \vec{r}_{i,j-1} - \vec{r}_{ij} \end{aligned}$$

are the successive coordinatewise vectors pointing away from the center point $\vec{r}_{ij}$ in the stencil of figure 1. Denoting the volume equalization and orthogonality sums by S_v and S_0 respectively, Kennon and Dulikravich minimized the linear combination

$$(1 - \alpha)S_v + \alpha S_0 \tag{95}$$

to generate or improve a grid. The parameter $0 \leq \alpha \leq 1$ was chosen to balance the two forces. The inclusion of adaptive forces comes with the inclusion of yet another sum. This corresponds with the basic pattern of most variational methods.

Variational Methods

The basic pattern of developments with variational methods is to gather the desired attributes, form a positive pointwise measure of each such attribute, integrate the measures over the field, form a positive linear combination of the integrals, and then minimize the resultant combination. The minimization process typically follows the standard manipulations from the calculus of variations (74). This leads to a system of partial differential equations known as Euler equations that are then to be solved by the available numerical methods for PDE's. The main attribute in most methods is the attraction to conformality and is given by the integrals of Eqs. 55 and 56 in the section on elliptic methods. This means that other attributes are balanced against the Laplace system for curvilinear variables that was employed by Winslow (245), Thompson, Thames and Mastin (215) and others. The main adaptive attribute typically comes in the form of a weighted Jacobian so that the minimized integral by itself would produce an equidistribution of the weight over volume elements or powers of them. Along with the attributes of conformality and weighted Jacobians, Yanenko et al (246) included a Lagrangian attraction for fluid motion while Brackbill and Saltzman (30) included an orthogonality attraction for an improved grid structure. The orthogonality attraction came from integrals of squared cross metrics. In two-dimensions, these were either $(g_{12})^2$ or $(g^{12})^2$. With the motivation from the use of orthogonality in Brackbill and Saltzman, Kennon and Dulikravich (129) pursued the finite volume approach discussed earlier, Saunders (179) examined a tensor product of B-splines, Kreis, Thames and Hassan (132) considered scaling problems, and Steinberg and Roache (207) developed a variant with reference grids.

Rather than employ the attraction to the preferred Laplace system for elliptic grid generation, Morice (150), Oskam and Huizing (160), and Steinberg and Roache (207) used attraction towards the Laplace system for locations in physical space. While this provides conformal smoothness, this is not as robust as the preferred system: it does however offer some simplicity. The main insurance against grid folding then falls upon the volume control or the permission of boundary point motion. Steinberg and Roache (207) favor volume control while Morice (150) and Oskam and Huizing (160) rely primarily upon boundary point motion.

The idea of reference grids employed by Steinberg and Roache (207) is the same as that employed earlier by Steger and Chaussee (204) in their development of hyperbolic methods. The use, however, is more extensive in that various properties are extracted from the reference grids. They appear essentially in the form of multiplicative factors applied to the standard format of the other investigators. These factors are weights that give appropriate ratios between the current desired grid and the known reference grid. A note of caution to be observed in the extraction of properties is that a desired attribute might not be obtained when the number of available degrees of freedom is exceeded.

A somewhat general development of the variational methods is given in the text by Thompson, Warsi, and Mastin (225). This was subsequently implemented in a large program. In one-dimension, earlier studies were given by Gough, Spiegel, and Toomre (105) and by Pierson and Kutler (163). Also, a study of temporal smoothness was developed by Bell and Shubin (17). In the multidimensional context, a study for systematically dealing with the inherent complexity of variational methods was performed by Steinberg and Roache (173,206) who proposed appropriate symbolic manipulation methods.

ADAPTIVE TRIANGULAR MESHES

In contrast to adaptive mesh strategies on curvilinear grids, general connectivity meshes offer simplicity and uniformity at the expense of a more complicated data structure. Dynamically resolving a flowfield is accomplished in one of two ways on both types of grids. Nodes can be added or deleted, or nodes can be displaced towards regions where resolution is required. Some schemes incorporate both options. Both node movement and grid enrichment (or depletion) become ill-suited to curvilinear grid structures as the degree of adaptivity increases. When moving nodes on a curvilinear grid, smoothness must be maintained for the coordinate transformation, and therefore, the node movement must be executed rather carefully. Furthermore, since the grid connectivity is fixed, some effort is required there to prevent the cells from distorting beyond the acceptable limit for a given numerical procedure. Rezoning procedures are a common remedy and permit a regridding of the domain, but are inherently dissipative. Modification of the total number of nodes on a structured grid presents one of two difficulties. If entire grid lines are simply added to the mesh, the increased resolution is extended into regions where it may not be needed, thereby unnecessarily increasing execution time and memory usage. A solution is to locally imbbed submeshes in regions where resolution is desired. Dangling grid lines at the boundaries typically arise and require specialized algorithms to be developed to properly transfer information between the fine and the coarser meshes (20). Conceptually, these difficulties are partially removed when the grid is unstructured. Once a library of low level routines is established to add, delete and move nodes, and to appropriately restructure the grid, adapting a triangular grid to a specified flowfield is relatively straightforward. Restructuring a triangular grid is a dynamic process which only involves local changes to the connectivity of the mesh. For example, the common edge of two adjacent triangles, viewed as the diagonal of the quadrilateral they form, can be removed and replaced by the quadrilateral's second diagonal according to some appropriate restructuring criteria. Simple restructuring schemes may also be dissipative if interpolation procedures are involved. This typically occurs when variables are stored at triangle centroids. However the grid is usually only affected in local regions. Restructuring occurs in regions where nodes are either added or subtracted. In the former case the resolution is sufficiently fine to remain unaffected by the dissipation, and in the latter, the structure of the flow is low enough to be unaffected.

Adaptive strategies on unstructured meshes have been pursued in several directions, the first of which comes from Lagrangian hydrodynamic simulations that are as free as possible from a rigid mesh structure. Free-Lagrangian methods, pioneered by the work of Crowley (42) on incompressible flows, represent the fluid by Lagrangian fluid particles which are carried with the convective velocity. These methods are particularly well-suited for problems with fluid interfaces, high shear regions, hydrodynamic instabilities, jet formation, and bubble collapse. The connectivity pattern between the particles is dynamically adjusted every several time steps to preserve a well-structured grid. As explained by (79,81), the optimal triangle shape based on standard truncation analysis is equilateral. As is the case with structured grids, flow variables can be stored at nodal points, cell centers or on cell edges.

A node-centered formulation permits grid restructuring without diffusion, since only triangle edges are reconnected while leaving the node positions fixed. Therefore the values of the flow variables remain unchanged. One disadvantage of this approach manifests itself when modeling material interfaces. In these cases it is not clear what interpretation to give interface nodes. This dilemma is removed in a zone-centered formulation. In this case, the restructuring algorithms must insure that edges remain aligned with the material interfaces. Although restructuring in regions of high flow gradients can introduce diffusion, these effects are localized to the restructured areas.

Among the restructuring algorithms proposed by Crowley are merging operations where two adjacent nodes are combined into a single node. While a complicated process in the context of pure triangular grids (since triangles are dynamically formed within a mesh containing quadrilaterals), it is suitable for Crowley's mixed cell approach. Problems with mixed cells not present in pure triangular meshes is that of bowtie or boomerang effects. These occur when the mesh movement creates non-convex quadrilaterals. Removal of these effects is accomplished through more sophisticated reconnection algorithms. Details of Crowley's reconnection algorithms are provided in (41). Fritts (88) has developed Free-Lagrangian algorithms on unstructured triangular grids, and has studied a wide variety of physical problems. Among others are the Kelvin-Helmoltz instability (84), the stability of free-surface waves (85), and combustion modeling (87). Three-dimensional Free-Lagrangian algorithms are still rather few. Trease (230) discusses detailed issues of Voronoi-cell based three-dimensional Free-Lagrangian methods. His code is an outgrowth of his 2-D Free-Lagrange code (229). A survey of two- and three- dimensional Free-Lagrangian algorithms is presented in (89).

An intermediate approach is adopted in the work of Erlebacher (79). The numerical scheme is Eulerian in the sense that the movement of the mesh points is unrelated to the convection velocity. Rather, the concept of monitor surface discussed above is a fundamental component of the algorithm. In its most basic form the strategy consists of assigning a positive mass density to each triangle. The mass density (also called weight) is a linear combination of operators acting on the monitor surface. For triangular grids, the weight function is defined by

$$w(x,y) = 1 + \alpha|\nabla f| + \beta\frac{|\nabla^2 f|}{2[1+(\nabla f)^2]^{3/2}}. \tag{96}$$

The term multiplied by β is an approximation to mean curvature. After multiplication of the mass density assigned to each individual triangle by its area, one obtains the equivalent of a sheet of variable mass. If a node c is surrounded by n nodes, old node positions $\vec{r}_i^{old}$ are updated by the center of mass formula

$$\vec{r}_c^{\,new} = \frac{\sum_{i=1}^{n} A_i w_i \vec{r}_i}{\sum_{i=1}^{n} A_i w_i} \tag{97}$$

where w_i labels the weight associated with the i^{th} triangle adjacent to node c. The area of triangle i is A_i. In comparison with the structured developments, this is the extension of Eq. 91. In the absence of curvature and gradient effects, it is clear that the node c becomes the centroid of the polygon formed by its adjacent nodes. As the weight becomes unevenly distributed, the movement is biased towards regions of greater pull. The algorithm is simple and readily vectorizable on today's supercomputers. Nodes may be added, but not subtracted. As with the Free-Lagrange algorithms, it is necessary to monitor the distortion of the grid and locally restructure it if necessary. Methods similar to those used by Fritts and Boris (86) have been adopted. Further discussion can be found in Erlebacher and Eiseman (80) and in Eiseman (70).

Adaptive methods in the context of finite-elements are also gaining in popularity. Miller et al. (146) and Miller (147) have developed specialized methods based on a finite-element formulation to treat sharp gradients and shocks. Briefly, node positions as well as function values are solved for in a unified finite-element treatment. A two dimensional Stefan problem was solved with this approach. A water-ice interface was followed in time, but very long and thin triangles were generated at the interface between two regions of relative uniformity. The fundamental deficiency was a lack of appropriate buffer regions to slow down the transition from very large triangles to very small ones.

Lõhner et. al. (134) have developed very efficient techniques to solve the steady state compressible Euler equations for high speed flows in two dimensions. Their strategy is to add and subtract nodes from the grid which is periodically refined. Mesh points remain fixed throughout the calculation. Specialized routines can distinguish between high gradients and shocks. This distinction is necessary to prevent unnecessary clustering along shock interfaces. If proper care is not taken, length scales decrease to zero in the presence of shock interfaces, and the standard resolution criteria is never satisfied. To increase the efficiency of their method, multigrid algorithms have been developed which are not based on a sequence of nested grids. Rather, the fine mesh is coarsened by regenerating the grid with a reduced number of nodes. Information is transferred from one grid to the next by a series of interpolation formulae (135). The algorithms are extremely robust and fully vectorized. With the use of hardware gather and scatter routines, solutions to complicated two-dimensional problems are obtained in 10-15 min. on a Cray-XMP.

Holmes and Lamson (121) solve the two-dimensional steady state Euler equations on an adaptive mesh. Grid refinement is solely accomplished via node addition. The transonic equations are solved adaptively by Palmerio and Dervieux (161). They use a spring analogy. Each edge of the grid is replaced by a spring whose stiffness is a function of the flow properties. Boundary nodes may be either moved or remain fixed. The resulting two-dimensional grids are well-structured, and can track shocks. However, special precautions must be taken to prevent the cell sizes from decreasing below a critical size due to strong flow structures.

Adaptive algorithms on unstructured grids are not restricted to triangular and mixed-cell type grids. Starting from a regular two-dimensional Cartesian mesh, Dannenhoffer and Baron (44) track the salient variables and refine individual cells as required to satisfy specified tolerance levels.

TEMPORAL FORMS FOR ADAPTIVITY

When the temporal accuracy of a simulation is to be enhanced, the evolving grid must closely follow the trajectories of the severe disturbances in the solution. This is in contrast to the situation where a steady state solution is sought and the grid there eventually settles down into a virtually final configuration. The primary concern in the development of an accurate temporal resolution is that the severe disturbances will not escape from their resolution during the course of any time step in the numerical simulation. With this concern in mind, a number of investigators have felt that the grid equations should be formulated directly for grid point velocities rather than positions. Then, at least, the grid velocities would follow an analytical model for any chosen time level such as the full implicit level or the level halfway between explicit and fully implicit. Clearly, the use of such velocities should be an improvement over using backward differences in time to estimate the same velocities from a grid point movement scheme that only produces pointwise locations from currently available data. From the viewpoint of grid velocities, the schemes which produce only pointwise locations can be referred to as static rather than dynamic (eg. note Hyman and Naughton (122)). A typical difference is that static methods depend upon data at only one instant of time while dynamic methods often depend upon data over an interval of time. With the advantage of directly obtaining grid velocities, the dynamic methods also possess the disadvantage of a more difficult control over coordinate nonsingularity. This is because grid point locations must ultimately be determined, and if some velocities are too large or change too quickly, then points could be either overrun or artificially encircled: correspondingly, there would be grid overlap or excessive skewness.

Among the dynamic grid velocity methods, there are methods proposed by Winkler, Mihalas and Norman (244), Bell and Shubin (17), Hindman, Kutler and Anderson (119), Hindman and Spencer (118), Rai and Anderson (166,167,3), Greenberg (106), Piva, DiCarlo, Favini, and Gui (164), Ghia, Ghia and Shin (99), and Harten and Hyman (109). Winkler, Mihalas, and Norman (244) develop a scheme based upon the equidistribution of weights over grid point indices. Nonsingularity in their one dimensional context is provided primarily by creating singularity barriers which kept the points from interchanging positions. This comes from equidistributing the cell eccentricity. In addition, they consider asymmetric time filtering to preserve resolution for the rapid reappearance of salient phenomena such as in shock wave reflections. The chief mechanism is a diffusion coefficient arising from a constructed factor on the time derivative. The factor contains enough residual memory to slow down the diffusion of the resolution just enough to allow for the rapid reappearance of small structures. Otherwise, resolution would have to rapidly disappear and then reappear, thus causing unnecessary numerical errors because of the temporal jerkiness. In comparison, Bell and Shubin (17) remove temporal jerkiness by balancing the weight function equidistribution against the magnitudes of time derivatives in the variational format. Their balancing coefficient was, however, only a constant and thus did not contain the residual memory as in the case of Winkler, Mihalas and Norman.

In another direction, Hindman and Spencer (118) converted the Poisson system of Eq. 63 into a grid velocity equation by formal temporal differentiation of the original equation. In addition, they also explored the use of equidistributed weight functions.

They found the same relationship to the forcing function in one-dimension that Anderson and Steinbrenner (5,6) eventually discovered in two-dimensions.

Identifications with alternate metaphors, such as the earlier spring analogy, have also provided the inspiration for several methods. In this spirit, Rai and Anderson (166,167) developed an analogy to a gravitational potential while Greenberg (106) related the grid movement to chemical reactions. The gravitational forces decayed according to an inverse power law for distances in the space of grid point indices. Force magnitudes and directions at each grid point came from the deviation of the error indicator from its average value. In a similar manner, the chemical reaction rates contained the adaptive data.

An even more direct use of physically based motivation occurred in the somewhat parallel studies of Piva et al. (164) and Ghia, Ghia and Shin (99). There, idealized two-dimensional momentum equations for viscous flow were transformed into diffusion equations. This was done because diffusion equations are easier to solve. The process basically amounts to a removal of the convective terms which would appear when the equations are expressed in terms of an arbitrary time dependent coordinate system. The two resulting equations are the grid movement equations which assume the Poisson format. In a sense, this is similar to the pursuit of Hindman and Spencer (118), although there is no consideration of equidistribution.

In addition to the static methods based upon the previous solution step and the dynamic grid velocity methods, there are methods which impose a grid distribution mechanism at some implicit level without the direct determination of a grid velocity. This includes the methods such as that employed by White (241) and that employed with the moving finite-element method investigated by Miller and Miller (146), Miller (147), Miller (148), Gelinas, Doss, and Miller (95), Herbst, Mitchell, and Schoobie (117), Baines (12), and Baines and Wathen (13). In the case of White (241) and others like it, the grid equation appears as a time-dependent constraint which is applied in an implicit coupled manner with the evolutionary physical equation. In the moving finite-element method, the coupling and the grid evolution comes directly out of the formal finite-element process when it is directly extended to include grid point motion.

In contrast to the dynamic or implicit temporal treatment of grid point motion, the static methods offer a great amount of simplicity, efficiency and spatial control at the expense of loosing the accurate tracking of severe disturbances. The corrective tracking measures taken are typically to either use a smaller time step or preferably to require a broader band of resolution. With the broader band, the idea is that the disturbance will still appear in the high resolution region at the end of the time step. Methods which lead to such breadth typically come from grid smoothness forces and from curvature clustering on the monitor surface.

The static methods also tend to offer more numerical stability for the class of problems where the broad banded resolution provides an adequate buffer for the containment of the disturbance. In the application, either the grid velocities are employed with backward differences in time or the solution is simply interpolated onto the new grid point locations in what is known as a remapping step. While both are commonly used, the remapping approach is more prevalent, particularly in cases where the steady state convergence is a prime element. In the steady state cases, the movement may start with direct interlacing between the PDE-solver and the grid generator and then progress to fewer and fewer applications of grid movement until movement is stopped altogether.

With trapezoidal finite-elements in one space dimension and time, Davis and Flaherty (46) and Flaherty et al. (82) employed a static grid generation scheme for a PDE-solver that was well-adjusted for temporal evolution. The static grid was generated from data at the explicit time level n for use at the implicit time level $n+1$. In a sense, this represents a zeroth order forward extrapolation of the grid in time. Because of the extrapolation, the tracking possibilities are absent. In contrast, Sanz-Serna and Christie (178) and Blom, Sanz-Serna and Verwer (26) consider a predictor step for the application of static grid generation. The essence of their idea is to apply the PDE-solver to get provisional solution values at $n+1$ and from those values to generate the grid at the implicit level $n+1$. Then with the grid determined at $n+1$, they get the actual solution at $n+1$. Although their development was one-dimensional in space, the idea of first predicting the implicit level adaptive data is attractive for any application in any number of spatial dimensions. The extra cost only amounts to an extra application of the PDE-solver. The benefits are the capability to accurately track rapidly moving severe disturbances by using static adaptive grid generators which are known to produce higher quality and more versatile grids.

CONCLUSION

In the course of our discussion, we have attempted to gain some perspective on the topic of grid generation. Rather than merely list and describe a sequence of topics in a loosely related fashion, we have stressed the various unifying concepts and have thereby hoped to highlight the whole topic by explaining the motivating factors behind the individual contributions and how these contributions came about.

The basic need for grid generation came from the geometric and topological complications that are common with the physical processes which occur in the design of many useful objects. The complications can arise either in the spatial region or in the physical properties. With the desire to numerically simulate such useful processes, grid generation then became essential.

To provide the general setting for grid generation, the fundamental topological issues were considered and were approached with various connectivity patterns for the points that discretely covered the field. In broad terms, these were split between structured and unstructured grids. Next, the principal grid generation techniques were discussed. These are split basically into algebraic methods and partial differential equation methods. In the application of the methods, the setting was seen to vary from completely automatic to interactive.

To address the problems with severe variations in a solution, the topic of adaptive grids was developed as a natural progression of the earlier methods. Although this was done for both structured and unstructured connectivity patterns, the emphasis was on the structured developments.

ACKNOWLEDGEMENT

The first author gratefully acknowledges the support given by both NASA and the Air Force under grants NAG3-715 and AFOSR-86-0307. The responsible people were Major. John P. Thomas and Dr. James D. Wilson at AFOSR and Bernard H. Anderson at NASA Lewis Research Center. In addition, support was also supplied by ICASE under NASA contract NAS1-18107.

REFERENCES

(1) Ablow, C. M., and Schecter, S., "Campolytropic Coordinates," Journal of Computational Physics, Vol. 27, 1978, pp. 351-362.

(2) Ablow, C.M., "Equidistant Mesh for Gas Dynamic Calculations," Numerical Grid Generation, Ed. by J. F. Thompson, 1982, pp. 859-863.

(3) Anderson, D. A., Rai, M. M., "The Use of Solution Adaptive Grids in Solving Partial Differential Equations," Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, pp. 317-338.

(4) Anderson, D. A., "Adaptive Grid Methods for Partial Differential Equations," Advances in Grid Generation, Ed. by K. N. Ghia and U. Ghia, 1983, pp. 1-15.

(5) Anderson, D. A. and Steinbrenner, J., "Generating Adaptive Grids with a Conventional Grid Scheme," AIAA-86-0427, 24th Aerospace Sciences Meeting, Reno, 1986.

(6) Anderson, D. A., "Constructing Adaptive Grids with Poisson Grid Generators," Numerical Grid Generation in Computational Fluid Dynamics, Ed. by J. Häuser and C. Taylor, Pineridge Press, 1986, pp. 125-136.

(7) Aston, M. B. and Thomas, J. W., "An Implicit Scheme for Water Wave Problems," Numerical Grid Generation, Ed. J.F. Thompson, North-Holland, 1982, pp. 809-818.

(8) Atta, E. H., "Component-Adaptive Grid Embedding," Numerical Grid Generation Techniques, Ed. by R. E. Smith, 1980, pp. 157-174.

(9) Atta, E. H. and Vadyak, J., "A Grid Interfacing Zonal Algorithm for Three-dimensional Transonic Flows about Aircraft Configurations," Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 170, Springer-Verlag, 1982, pp. 107-114, (Also AIAA Journal, vol. 21, 1983, pp. 1271-1277).

(10) Augenbaum, J. M., "A Lagrangian Method Based on the Voronoi Diagram for the Incompressible Navier-Stokes Equations on a Periodic Domain," The Free-Lagrange Method, Lecture Notes in Physics, Springer-Verlag, New York, 1985, pp. 236-254.

(11) Babuška I., Chandra, J. and Flaherty, J. E. (Eds.) Adaptive Computational Methods for Partial Differential Equations, SIAM 1983.

(12) Baines, M. J., "On Approximate Solutions of Time-Dependent Partial Differential Equations by the Moving Finite-Element Method," Numerical Analysis Report 1/85, University of Reading, U. K., Numerical Methods in Fluid Dynamics II, Ed. by K. W. Morton and M. J. Baines, Oxford University Press, 1986.

(13) Baines, M. J. and Wathen, A. J., "Moving Finite-Element Modeling of Compressible Flows," Applied Numerical Mathematics, No. 2, 1986, pp. 495-514.

(14) Baker, T. J., "Mesh Generation by a Sequence of Transformations," Journal of Applied Numerical Mathematics, Vol. 2, No. 6, 1986, pp. 515-528.

(15) Bank, R. E., "The Efficient Implementation of Local Mesh Refinement Algorithms," Adaptive Computational Methods for Partial Differential Equations, Ed. by I. Babuška, J. Chandra, and J. E. Flaherty, SIAM, Philadelphia, 1983, pp. 74-81.

(16) Barger, R. L. and Adams, M. S., "Semianalytic Modeling of Aerodynamic Shapes," NASA Technical Paper 2413, 1985.

(17) Bell, J. B. and Shubin, G. R., "An Adaptive Grid Finite Difference Method for Conservation Laws," Journal of Computational Physics, Vol. 52, 1983, pp. 569-591.

(18) Benek, J. A., Steger, J. L. and Dougherty, F. C., "A Flexible Grid Embedding Technique with Application to the Euler Equations," Proceedings of the 6th AIAA Computational Fluid Dynamics Conference, Danvers, MA, 1983, pp. 373-382.

(19) Benek, J. A. and Buning, P. G. and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," Proceedings of the 7th AIAA Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 322-331.

(20) Berger, M. J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," Report No. STAN-CS-82-924, Ph.D. Thesis, Stanford University, 1982, pp. 1-118.

(21) Berger, M. J. and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations," AIAA Journal, Vol. 23, No. 4, 1985, pp. 561-568.

(22) Berger, J. J., "On Conservation at Grid Interfaces," ICASE Report No. 84-43, 1984, also in SIAM Numerical Analysis, to appear 1987.

(23) Berger, M. J. and Oliger, J., "Adaptive Mesh Refinement for Hyperbolic PDE's," Journal of Computational Physics, Vol. 53, 1984, pp. 484-512.

(24) Berglind, T., "The Generation of Continuous Multi-Block Grids Around Car Configurations above a Flat Ground Plane using Transfinite Interpolation," Numerical Grid Generation in Computational Fluid Dynamics, Ed. Hãuser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(25) Birkhoff, G. and Gordon, W. J., "The Draftsman's and Related Equations," Journal of Approximation Theory, Vol. 1, 1968, pp. 191-208.

(26) Blom, J. G., Sanz-Serna, J. M. and Verwer, J. G., "On Simple Moving Grid Methods for One-dimensional Evolutionary Partial Differential Equations," Centrum voor Wiskunder end Informatica, Department of Numerical Mathematics, Report NM-R8620, 1986.

(27) Boerstoel, J. W., "Problem and Solution Formulations for the Generation of 3D Block-Structured Grids," Numerical Grid Generation in Computational Fluid Dynamics, Ed. Hãuser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(28) Bõrgers, C. and Peskin, C. S., "A Lagrangian Method Based on the Voronoi Diagram for the Incompressible Navier-Stokes Equations on a Periodic Domain," The Free-Lagrange Method, Ed. by M. J. Fritts, W. P. Crowley, and H. Trease, Lecture Notes in Physics, Vol. 238, Springer-Verlag, 1985, pp. 87-113.

(29) Brackbill, J. U. and Saltzman, J., "An Adaptive Computation Mesh for the Solution fo Singular Perturbation Problems," Numerical Grid Generation Techniques, Ed. by R. E. Smith, [illegible], pp. 193-196.

(30) Brackbill, J. U. and Saltzman, J. S., "Adaptive Zoning for Singular Problems in Two Dimensions," Journal of Computational Physics, Vol. 46, 1982, p. 342.

(31) Brackbill, J. U., "Coordinate System Control: Adaptive Meshes," Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, pp. 277-294.

(32) Brown, D., "Solution Adaptive Mesh Procedure for the Numerical Solution of Singular Perturbation Problems," California Institute of Technology, Ph.D Thesis, Applied Mathematics, 1982.

(33) Carcaillet, R., Dulikravich, G. S., and Kennon, S. R., "Generation of Solution Adaptive Computational Grids using Optimization," Computer Methods in Applied Mechanics and Engineering, Vol. 57, 1986, pp. 279-295.

(34) Carcaillet, R., Kennon, S. R., and Dulikravich, "Optimization of Three-Dimensional Grids," AIAA Journal of Aircraft, Vol. 23, No. 5, 1986 pp. 415-421.

(35) Chakravarthy, S. and Anderson, D. A , "Numerical Conformal Mapping," Mathematics of Computation, Vol. 33, 1979, pp. 953-969.

(36) Chern, I-L., Glimm, J., McBryan, O., Plohr, B., Yaniv, S., "Front Tracking for Gas Dynamics," Journal of Computational Physics, Vol. 62, 1986, pp. 83-110.

(37) Chesshire, J., "Composite Grid Construction and Applications," Ph.d. Thesis, Applied Math. California Institute of Technology, 1986.

(38) Coleman, R. M., "Generation of Boundary-Fitted Coordinate Systems Using Segmented Computational Regions," Numerical Grid Generation, Ed. by J. F. Thompson, North Holland, 1982, pp. 663-652.

(39) Coleman, R. M. and Brabanski, M. L., "Numerical Grid Generation for Three-Dimensional Geometries Using Segmented Computational Regions," Numerical Grid Generation in Computational Fluid Dynamics, Ed. by Hãser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(40) Connett, W. C., Agarwal, R. K., and Schwartz, A. L., "An Adaptive Grid Generation Scheme for Flow Field Calculations," AIAA-87-0199, 25th Aerospace Sciences Meeting, Reno, 1987.

(41) Cooper, R. E., "Data Structures and Vectorization in a Two-Dimensional Free-Lagrangian Code," The Free-Lagrange Method, Ed. by M. J. Fritts, W. P. Crowley, and H. Trease, Lecture Notes in Physics, Vol. 238, Springer-Verlag, 1985, pp. 182-192.

(42) Crowley, W. P., "FLAG: A Free-Lagrange Method for Numerically Simulating Hydrodynamic Flows in Two Dimensions," Proceedings of the 2nd International Conference on Numerical Methods in Fluid Dynamics, 1970.

(43) Crowley, W. P., "Free-Lagrange Methods for Compressible Hydrodynamics in two Space Dimensions," The Free-Lagrange Method, Ed. by M. J. Fritts, W. P. Crowley, and H. Trease, Lecture Notes in Physics, Vol. 238, Springer-Verlag, 1985, pp. 1-21.

(44) Dannenhoffer III, J. F. and Baron, J. R., "Adaptive Procedures for Steady State Solution of Hyperbolic Equations," AIAA-84-005, 22nd Aerospace Sciences Meeting, Reno, 1984.

(45) Davis, R. ,T., "Numerical Methods for Coordinate Generation Based on Schwarz-Christoffel Transformation," Paper No. 79-1463, Proceedings of the AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Va., 1979.

(46) Davis, S. F. and Flaherty, J. E., "An Adaptive Finite-Element Method for Initial-Value Problems for Partial Differential Equations," SIAM Journal of Scientific and Statistical Computing, Vol. 3, 1982, pp. 6-27.

(47) DiCarlo, A., Piva, R., Guj, G., "A Study on Curvilinear Coordinates and Macro-Elements for Multiply Connected Flow Fields," Proceedings of the 6th International Conference on Numerical Methods in Fluid Dynamics, Ed. by H. Cabannes, M. Holt and V. Rusanov, Lecture Notes in Physics, Vol. 90, springer-Verlag, 1978, pp. 161-168.

(48) Diaz, A. R., Kikuchi, N. and Taylor, J. E., "A Method of Grid Optimization for Finite-Element Methods," Computational Methods in Applied Mechanics and Engineering, Vol. 41, 1983, pp. 29-45.

(49) Dukowicz, J., "Lagrangian Fluid Dynamics Using the Voronoi-Delaunay Mesh," in Numerical Methods for Coupled Problems, Pineridge Press, Swansea, U.K., 1981.

(50) Dwyer, H. A., Kee, R. J., and Sanders, B. R., "An Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," Proceedings of the AIAA Computational Fluid Dynamic Conference, Ed. by E. Murman, Williamsburg, Virgina, 1979, pp. 195-203, also in AIAA Journal, Vol. 18, 1980, pp. 205-212.

(51) Dwyer, H. A., Smooke, M. D., and Kee, R. J., "Adaptive Gridding for Finite Difference Solutions to Heat and Mass Transfer Problems," Numerical Grid Generation, Ed. by J. F. Thompson, North Holland, 1982, pp. 339-356.

(52) Dwyer, H. A., "A Discussion of Some Criteria for the Use of Adaptive Gridding," Adaptive Computational Methods for Partial Differential Equations, Ed. by I. Babuška, J. Chandra, and J. E. Flaherty, SIAM, 1983, pp. 111-122.

(53) Dwyer, H. A., "Grid Adaption for Problems in Fluid Dynamics," AIAA Journal, Vol. 22, No. 12, 1984, pp. 1705-1712.

(54) Dwyer, H. A., and Onyejekwe, O. O., "Generation of Fully Adaptive and/or Orthogonal Grids," Proceedings of the 9th International Conference on Numerical Methods in Fluid Dynamics, Ed. by Soubbaramayer and Boujot, Saclay, France, Lecture Notes in Physics 218, Springer-Verlag, 1985, pp. 203-207.

(55) Eberhardt, S. and Baganoff, D., "Overset Grids in Compressible Flow" Proceedings of the 7th AIAA Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 332-339.

(56) Edwards, T. A., "Noniterative Three-Dimensional Grid Generation Using Parabolic Partial Differential Equations," AIAA-85-0485, 23rd Aerospace Sciences Meeting, 1985.

(57) Eiseman, P. R., "Adaptive Grid Generation," Computer Methods in Applied Mechanics and Engineering, to appear.

(58) Eiseman, P. R., "The Numerical Solution of the Fluid Dynamical Equations in Curvilinear Coordinates," AFWL-TR-73-172, Air-Force Weapons Laboratory, 1973, pp. 1-85.

(59) Eiseman, P. R., "A Coordinate System for a Viscous Transonic Cascade Analysis," Journal of Computational Physics, Vol. 26, No. 3, 1978, pp. 307-338.

(60) Eiseman, P. R., "A Multi-Surface Method of Coordinate Generation," Journal of Computational Physics, Vol. 33, 1979, pp. 118-150.

(61) Eiseman, P. R. and Smith, R., "Mesh Generation Using Algebraic Techniques," Numerical Grid Generation Techniques, Ed. by R. E. Smith, NASA CP-2166, 1980, pp. 73-120.

(62) Eiseman, P. R., Geometric Methods in Computational Fluid Dynamics, ICASE Report 80-11 and von Karman Institute for Fluid Dynamics Lecture Series Notes, 1980.

(63) Eiseman, P. R., "Coordinate Generation with Precise Controls over Mesh Properties," Journal of Computational Physics, Vol. 47, 1982, pp. 331-351.

(64) Eiseman, P. R., "High Level Continuity for Coordinate Generation with Precise Controls," Journal of Computational Physics, Vol. 47, 1982, pp. 352-374.

(65) Eiseman, P. R., "Orthogonal Grid Generation," Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, pp. 193-233.

(66) Eiseman, P. R., "Automatic Algebraic Coordinate Generation," Numerical Grid Generation, Ed. by J. F. Thompson, North Holland, 1982, pp. 447.

(67) Eiseman, P. R., "Alternating Direction Adaptive Grid Generation," Proceedings of the 6th AIAA Computational Fluid Dynamics Conference, Danvers, MA, 1983, pp. 339-348. (Also in AIAA Journal, Vol. 23, No. 4, 1985, pp. 551-560).

(68) Eiseman, P. R., "Adaptive Grid Generation by Mean Value Relaxation," Advances in Grid Generation, Ed. by K. N. Ghia, and U. Ghia, ASME-FED, Vol. 5, 1983, pp. 29-34, also in ASME Journal of Fluids Engineering, Vol. 107, 1985, pp. 477-483.

(69) Eiseman, P. R., "Grid Generation for Fluid Mechanics Computations," Annual Review of Fluid Mechanics, Vol 17, 1985, pp. 487-522.

(70) Eiseman, P. R., "Solution Adaptivity using a Triangular Mesh," The Free-Lagrange Method, Lecture Notes in Physics, Vol. 238, Springer-Verlag, New York, 1985, pp. 206-235.

(71) Eiseman, P. R., "The Geometrical Construction of Pointwise Distributions on Curves," Applied Mathematics and Computations, Vol. 21, No. 3, Part 2, 1987, pp. 233-276.

(72) Eiseman, P. R., "Alternating Direction Adaptive Grid Generation for Three-Dimensional Regions," Proceedings of the 10th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 264, Springer-Verlag, 1986, pp. 258-263.

(73) Eiseman, P. R., "A Control Point Form of Algebraic Grid Generation," 5th International Conference on Numerical Methods in Laminar and Turbulent Flow, July 6-10th 1987," Pineridge Press, C. Taylor, W.G. Habashi and M.M. Hafez (eds.), 1987, 1083-1091.

(74) Elsgolc, L. E., Calculus of Variations, Pergamon Press (London) and Addison-Wesley (Reading, MA), 1962.

(75) Epstein, B., Partial Differential Equations: An Introduction, McGraw-Hill, 1962.

(76) Eriksson, L.-E., "Flow Solution on a Dual-Block Grid Around an Airplane," Computer Methods in Applied Mechanics and Engineering, to appear.

(77) Eriksson, L. E., "Generation of Boundary-Conforming Grids around Wing-Body Configurations Using Transfinite Interpolation," AIAA Journal, Vol. 20, No. 10, 1982, pp. 1313-1320.

(78) Eriksson, L.-E., "A Study of Mesh Singularities and their Effects on Numerical Errors," FFA Technical Note TN 1984-10, The Aeronautical Research Institute of Sweden, 1984.

(79) Erlebacher, G., "Solution Adaptive Triangular Meshes with Application to Plasma Equilibrium," Doctoral Thesis, Department of Applied Physics and Nuclear Engineering, Columbia University, 1984.

(80) Erlebacher, G. and Eiseman, P. R., "Adaptive Triangular Mesh Generation," AIAA-84-1607, 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, 1984, also in AIAA Journal, to appear.

(81) Erlebacher, G., "Finite Difference Operators on Unstructured Triangular Meshes," The Free-Lagrange Method, Ed. by M. J. Fritts, W. P. Crowley, and H. Trease, Lecture Notes in Physics, Vol. 238, Springer-Verlag, 1985, pp. 22-53.

(82) Flaherty, J. E., Coyle, J. J., Ludwig, R. and Davis, S. F., "Adaptive Finite-Element Methods for Parabolic Partial Differential Equations," Adaptive Computational Methods for Partial Differential Equations, Ed. by I. Babuška, J. Chandra, and J. E. Flaherty, SIAM, Philadelphia, 1983, pp. 144-164.

(83) Forsey, C. R., Edwards, M. G., and Carr, M. P., "An Investigation into Grid Patching Techniques," Numerical Grid Generation Techniques, Ed. by R. E. Smith, 1980, pp. 265-293.

(84) Fritts, M. J., "Lagrangian Simulations of the Kelvin-Helmholtz Instability," SAI Report SAI-76-632-WA, 1976.

(85) Fritts, M. J., "A Numerical Study of Free-Surface Waves," SAI Report SAI-76-528-WA, 1976.

(86) Fritts, M. J. and Boris, J. P., "The Lagrangian Solution to Transient Problems in Hydrodynamics using a Triangular Mesh", Journal of Computational Physics, Vol. 31, 1979, pp. 173-215.

(87) Fritts, M. J., Oran, E. S., and Boris, J. P., "Lagrangian Fluid Dynamics for Combustion Modeling," NRL Memorandum Report 4570, 1981.

(88) Fritts, M., "Three-Dimensional Algorithms for Grid Restructuring in Free-Lagrangian Calculations," The Free-Lagrange Method, Ed. by M. J. Fritts, W. P. Crowley, and H. Trease, Lecture Notes in Physics, Vol. 238, Springer-Verlag, 1985, pp. 122-144.

(89) Fritts, M. J., Crowley, W. P. and Trease, H. (Eds.) The Free-Lagrange Method, Lecture Notes in Physics, Vol. 238, Springer-Verlag, 1985.

(90) Fritz, W., "Numerical Grid Generation around Complete Aircraft Configurations," AGARD, 58th Metting of Applications of Computational Fluid Dynamics in Aeronautics," Aix-en-Provence, France, 1986.

(91) Fritz, W. and Leicher, S., "Numerical Solution of 3-D Inviscid Flow Fields around Complete Aircraft Configurations," 15th Congress International council of the Aeronautical Sciences, London, UK, ICAS-86-1.3.2, 1986.

(92) Fuchs, L., "A Zonal Multigrid Method for the Simulation of Flows in a System of Channels," Numerical Grid Generation in Computational Fluid Dynamics, Ed. by J. Häuser and C. Taylor, Pineridge Press, 1986, pp. 463-472.

(93) Gaitonde, D., "Three-Dimensional Interactive Grid Generation," Doctoral Thesis, Rutgers University, 1986, pp. 1-142.

(94) Garon, A. and Camarero, R., "Generation of Surface Fitted Grids," Advances in Grid Generation, Ed. by Ghia and Ghia, ASME-FED, Vol. 5, 1983, pp. 117-122.

(95) Gelinas, R., Doss, S., and Miller, K., "The Moving Finite-Element Method: Applications to General Equations with Multiple Large Gradients," Journal of Computational Physics, Vol. 40, 1981, pp. 202-249.

(96) Ghia, U., Ghia, K. N., and Studerus, C. J., "Use of Surface-Oriented Coordinates in the Numerical Simulation of Flow in a Turbine Cascade," Proceedings of the 5th International Conference on Numerical Methods in Fluid Dynamics, Ed. by A.I. van de Vooren and P. J. Zandbergen, Lecture Notes in Physics, Vol. 59, Springer-Verlag, 1976, pp. 197-204.

(97) Ghia, K. N. and Ghia, U. (Eds.), Advances in Grid Generation, FED-Vol. 5, ASME Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, 1983.

(98) Ghia, U., Ghia, K. N., and Ramamurti, R., "Hybrid C-H Grids for Turbomachinery Cascades," Advances in Grid Generation, Ed by K. N. Ghia and U. Ghia, ASME, Vol. 5, 1983, pp. 143-149.

(99) Ghia, K., Ghia, U., Shin, C. T., "Adaptive Grid Generation for Flows with Local High Gradient Regions," Advances in Grid Generation, Ed. by Ghia and Ghia, ASME-FED, Vol. 5, 1983, pp. 35-48.

(100) Glimm, J., McBryan, O., Menikoff, R., and Shang, D. H., "Front Tracking Applied to Rayleigh-Taylor Instability," SIAM Journal of Scientific and Statistical Computing, Vol. 7, No. 1, 1986, pp. 230-251.

(101) Gnoffo, P. A., "A Vectorized, Finite Volume, Adaptive-Grid Algorithm for Navier-Stokes," Numerical Grid Generation, Ed. by J. F. Thompson, 1982, pp. 819-835.

(102) Gordon, W. J., "Blending-Function Methods of Bivariate and Multivariate Interpolation and Approximation," SIAM Journal of Numerical Analysis, Vol. 8, 1971, pp. 158-177.

(103) Gordon, W. J., Hall, C. A., "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation," International Journal of Numerical Methods in Engineering, Vol. 7, 1973, pp. 461-477.

(104) Gordon, W. J., Thiel, L. C., "Transfinite Mappings and their Application to Grid Generation," Numerical Grid Generation, 1st Ed., New York, North Holland, 1982, pp. 171-192.

(105) Gough, D. O., Spiegel, E. A., and Toomre, J., "Highly Stretched Mesh as functionals of Solutions," Proceedings of the 4th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 35, Springer-Verlag, 1975, p. 191.

(106) Greenberg, J. B., "A New Self-Adaptive Grid Method," Proceedings of the 6th AIAA Computational Fluid Dynamics Conference, Danvers, MA, 1983, pp. 323-332, also in AIAA Journal, Vol. 23, No. 2, 1985, pp. 317-320).

(107) Halsey, D., "Conformal Grid Generation for Multi Element Airfoils," Numerical Grid Generation, Ed. by J. F. Thompson, North Holland, 1982, pp. 585-600.

(108) Halsey, D., "Calculation of Compressible Potential Flow about Multi Element Airfoils using a Source Field-Panel Approach," AIAA 23rd Aerospace Sciences Meeting, 1985, paper AIAA-85-0038, 1985.

(109) Harten, A. and Hyman, J. M., "A Self-Adjusting Grid for the Computation of Weak Solutions of Hyperbolic Conservation Laws," Journal of Computational Physics, Vol. 50, 1983, pp. 235-269.

(110) Hāuser, J., Papp, H. G., Eppel, D. and Sengupta, S., "Boundary Conformed Coordinate Systems for Selected Two-dimensional Fluid flow Problems, Part I: Application of the BFG Method," International Journal for Numerical Methods in Fluids, Vol. 6, 1986, pp. 507-527.

(111) Hāuser, J. and Taylor, C. (Eds.), Numerical Grid Generation in Computational Fluid Dynamics, Pineridge Press, 1986.

(112) Haussling, H. J., "Body-Fitted Coordinates for Accurate Numerical Solutions of Multi-Body Flow Problems," Journal of Computational Physics, Vol. 30, 1979, pp. 107-124.

(113) Haussling, H. J. and Coleman, R. M., "Nonlinear Water Waves Generated by an Accelerated Circular Cylinder," Journal of Fluid Mechanics, Vol. 92, 1979, p. 267.

(114) Haussling, H. J., "Solution of Nonlinear Water Wave Problems using Boundary-Fitted Coordinate Systems," Numerical Grid Generation, Ed. J.F. Thompson, North-Holland, 1982, pp. 385-407.

(115) Henshaw, W. D. and Chesshire, J., "Multigrid on Composite Meshes," SIAM Scientific and Statistical Computing, to appear.

(116) Henshaw, W. D., "Part I: Numerical Solution of Hyperbolic Systems of Conservation laws, Part II: Composite Overlapping Grid Techniques," Ph.D. Thesis, Applied Math., California Institute of Technology, 1985.

(117) Herbst, B., Mitchell, A. and Schoobmie, S., "A Moving Petrov-Galerkin Method for Transport Equations," International Journal for Numerical Methods in Engineering, Vol. 18, 1982, pp. 1321-1336.

(118) Hindman, R. G. and Spencer, J., "A New Approach to Truly Adaptive Grid Generation," AIAA 21st Aerospace Sciences Meeting, Reno, 1983, Paper AIAA-83-0450.

(119) Hindman, R.G., Kutler, P., and Anderson, D., "Two-Dimensional Unsteady Euler Equation solver for Arbitrarily Shaped Flow Regions," AIAA Journal, Vol. 19, No. 4, pp. 424-431.

(120) Hodge, J. K., Leone, S. A., and McCarty, R. L., "Non-Iterative Parabolic Grid Generation for Parabolized Equations," Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 371-380.

(121) Holmes, D. G. and S. H. Lamson, "Adaptive Triangular Meshes for Compressible Flow Solutions," Numerical Grid Generation in Computational Fluid Dynamics, Ed. Hāuser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(122) Hyman, J. M. and Naughton, M. J., "Static Rezone Methods for Tensor Product Grids," Lectures in Applied Mathematics, Vol. 22, American Mathematical Society, 1985, pp. 321-343.

(123) Ives, D. C. and Menor, W. A., "Grid Generation for Inlet and Inlet-Center body Configurations Using Conformal Mapping and Stretching," Paper No. 81-0997, Proceedings of the AIAA 5th Computational Fluid Dynamics Conference, 1981.

(124) Ives, D. C. , "Conformal Grid Generation," Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, pp. 107-135.

(125) Ives, D. C. and Liutermoza, J. F., "Analysis of Transonic Cascade Flow Using Conformal Mapping and Relaxation Techniques," AIAA Journal, Vol. 15, 1977, pp. 647-642.

(126) Jameson, A., Baker, T. J., Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft," AIAA-86-0103, Aerospace Sciences Meeting, Reno, 1986.

(127) Kellogg, O. D., Foundations of Potential Theory, Dover, 1953.

(128) Kennon, S. R. and Dulikravich G. S., "Composite Computational Grid Generation Using Optimization," Numerical Grid Generation in Computational Fluid Dynamics, Ed. Hãuser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986, pp. 217-225.

(129) Kennon, S. R. and Dulikravich, G. S., "Generation of Computational Grids Using Optimization," AIAA Journal, Vol. 24, No. 7, 1986, pp. 1069-1073.

(130) Kober, H., Dictionary of Conformal Representations, Dover Publications, Inc., New York, 1957.

(131) Kowalski, E. J., "Boundary-Fitted Coordinate Systems for Arbitrary Computational Regions," Numerical Grid Generation Techniques, Ed. by R. E. Smith, 1980, pp. 331-353.

(132) Kreis, R. I., Thames, F. C., and Hassan, H. A., "Application of a Variational Method for Generating Adaptive Grids," AIAA Journal, Vol. 24, No. 3, 1985, pp. 404-410.

(133) Kreiss, B., "Construction of a Curvilinear Grid," SIAM Journal on Scientific and Statistical Computing, Vol. 4, No. 2, 1983, pp. 270-279.

(134) Lõhner, R., Morgan, K., Peraire, J. and Zienkiewicz, O. C., "Recent Developments in FEM-CFD," The Free-Lagrange Method, Lecture Notes in Physics, Vol. 238, Springer-Verlag, New York, 1985, pp. 236-254.

(135) Lõhner, R., Morgan, K., "An Unstructured Multigrid Method for Elliptic Problems," International Journal of Numerical Methods in Engineering, Vol. 24, 1987, pp. 101-115.

(136) Lombard, C. K. and Venkatapathy, E., "Implicit Boundary Treatment for Joined and Disjoint Patched Mesh Systems," Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 129-140.

(137) Marshall, G., Eiseman, P. R. and Kuo, J. T., "A General Collapsing Technique for Three-Dimensional Algebraic Grid Generation," Journal of Computational Physics, Vol. 62, No. 1, 1986, pp. 180-198.

(138) Mastin, C. W., Thompson, J. F., "Elliptic Systems and Numerical Transformations," Journal of Mathematical Analysis and Applications, Vol. 62, 1978, pp. 52-62.

(139) Mastin, C. W. and Thompson, J. F., "Transformation of Three-Dimensional Regions onto Rectangular Regions by Elliptic Systems," Numerische Mathematik, Vol. 29, 1978, pp. 397-407.

(140) Mastin, C. W. and McConnaughey, H. V., "Computational Problems on Composite Grids," AIAA-84-1611, 1984.

(141) Mastin, C. W., "Interface Procedures for Overlapping Grids," Numerical Grid Generation in Computational Fluid Dynamics, Ed. by J. Häuser and C. Taylor, Pineridge Press, 1986, pp. 227-245.

(142) Mavriplis, D. and Jameson, A., "A Multigrid Solution of the Two-Dimensional Euler Equations on Unstructed Triangular Meshes," AIAA paper AIAA-87-0353, AIAA 25th Aerospace Sciences Meeting, 1987.

(143) Middlecoff, J. F., Thomas, P. D., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," Proceedings of the AIAA Computational Fluid Dynamic Conference, 4th, Williamsburg, Virgina, 1979, pp. 175-179.

(144) Miki, K. and Takagi, T., "A Domain Decomposition and Overlapping Method for the Generation of Three-Dimensional Boundary-Fitted Coordinate Systems," Journal of Computational Physics, Vol. 53, pp. 319, 1984.

(145) Miki, K. and Takagi, T., "Numerical Solution of Poisson's Equation with Arbitrarily Shaped Boundaries using a Domain Decomposition and Overlapping Technique, Journal of Computational Physics, Vol. 67, No. 2, 1986, pp. 263-278.

(146) Miller, K. and Miller, R. N., "Moving Finite Elements. I," SIAM Journal of Numerical Analysis, Vol. 18, No. 6, 1981, pp. 1019-1032.

(147) Miller, K., "Moving Finite Elements. II," SIAM Journal of Numerical Analysis, Vol. 18, No. 6, 1981, pp. 1033-1057.

(148) Miller, K, "Alternate Modes to Control the Nodes in the Moving Finite Element Method," Adaptive Computational Methods for Partial Differential Equations, Ed. by I. Babuška, J. Chandra, and J. E. Flaherty, SIAM, Philadelphia, 1983, pp. 165-182.

(149) Moretti, G., "Grid Generation using Classical Techniques," Numerical Grid Generation Techniques, Ed. by R. E. Smith, NASA CP2166, 1980, pp. 1-36.

(150) Morice, P., "Numerical Generation of Boundary-Fitted Coordinate Systems with Optimal Control of Orthogonality," Advances in Grid Generation, Ed. by K. N. Ghia, and U. Ghia, ASME-FED-Vol. 5, 1983, pp. 29-34, also in ASME Journal of Fluids Engineering, Vol. 107, 1985, pp. 71-78, also ONERA-TP-082, 1983.

(151) Nakahashi, K. and Deiwert, G. S., "A Practical Adaptive Grid Method for Complex Fluid Flow Problems," Proceedings of the 9th International Conference on Numerical Methods in Fluid Dynamics, Ed. by Soubbaramayer and Boujot, Saclay, France, Lecture Notes in Physics 218, Springer-Verlag, 1985, pp. 422-426.

(152) Nakahashi, K. and Deiwert, G. S., "A Self-Adaptive Grid Method with Application to Airfoil Flow," Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 340-350.

(153) Nakahashi, K., "FDM-FEM Zonal Approach for Computations of Computations of Compressible Viscous Flows," Proceedings of the 10th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 264, Springer-Verlag, 1986, pp. 484-498.

(154) Nakahashi, K. and Deiwert, G. S., "Three-Dimensional Adaptive Grid Method," AIAA Journal, Vol. 24, No. 6, 1986, pp. 948-954.

(155) Nakamura, S., "Marching Grid Generation Using Parabolic Partial Differential Equations," Numerical Grid Generation, 1st Ed., New York, North Holland, 1982, pp. 775-786.

(156) Nakamura, S., "Noniterative Grid Generation Using Parabolic Difference Equations for Fuselage-Wing Flow Calculations," Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 170, Springer-Verlag, 1982, pp. 396-401.

(157) Oden, J. T. and Reddy, J. N., An Introduction to the Mathematical Theory of Finite Elements," Wiley, 1976.

(158) Oden, J. T. , Devloo, P. and Strouboulis, T., "Adaptive Finite-Element Methods for the Analysis of Inviscid Compressible Flow: I. Fast Refinement/Unrefinement and Moving Mesh Methods for Unstructured Meshes," to appear in Computational Methods in Applied Mechanics and Engineering.

(159) Oliger, J., "Adaptive Grid Methods for Hyperbolic Partial Differential Equations," Inverse Problems of Acoustic and Elastic Waves, Ed. by F. Santosa, Y. H. Pao, W. W. Symes, and C. Holland, SIAM, Philadelphia, 9184, pp. 320-331.

(160) Oskam, B. and Huizing, G. H., "Flexible Grid Generation for Complex Geometries in two Space dimensions Based on Variational Principles," National Aerospace Laboratory NLR, Amsterdam, The Netherlands, 1986, Report NLR MP86024L.

(161) Palmerio, B. and Dervieux, A., "Application of a FEM Moving Node Adaptive Method to Accurate Shock Capturing," Numerical Grid Generation in Computational Fluid Dynamics, Ed. Hãuser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(162) Pennisi, L. L., Elements of Complex Variables, Host, Rinehart and Winston, New York, 1963.

(163) Pierson, B. L. and Kutler, P., "Optimal Nodal Point Distribution for Improved Accuracy in Computational Fluid Dynamics," AIAA Journal, Vol. 18, No. 1, 1980, pp. 49-54.

(164) Piva, R., DiCarlo, A., Favini, B., Guj, G., "Adaptive Curvilinear Grids for Large Reynolds Number Viscous Flows," Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 170, Springer-Verlag, 1982, pp. 414-419.

(165) Protter, M. H. and Weinberger, H. F., Maximum Principles in Differential Equations, Prentice-Hall, 1967.

(166) Rai, M. M. and Anderson, D. A., "Grid Evolution in Time Asymptotic Problems," Numerical Grid Generation Techniques, Ed. by R. E. Smith, 1980, pp. 409-430, also in Journal of Computational Physics, Vol. 43, 1981, pp. 327-344.

(167) Rai, M. M. and Anderson, D. A., "Application of Adaptive Grids to Fluid Flow Problems with Asymptotic Solutions," AIAA Journal, Vol. 20, 1982, pp. 496-502.

(168) Rai, M. M., "A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations," AIAA-84-0164, Aerospace Sciences Meeting, Reno, 1984.

(169) Rai, M. M., "Navier-Stokes Simulations of Rotor-Stator Interaction Using Patched and Overlaid Grids," Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 282-298.

(170) Rai, M. M., "An Implicit Conservative Zonal Boundary Scheme for Euler Equation Calculations," AIAA-85-0488, Aerospace Sciences Meeting, Reno, 1985.

(171) Rai, M. M., "A Relaxation Approach to Patched-Grid Calculations with the Euler Equations," Journal of Computational Physics, Vol. 66, No. 1, 1986,

(172) Reyna, L., "Part I: Stability of Chebyshev Collocation, Part II: Interpolation for Surfaces with 1-D Discontinuities, Part III: On Composite Meshes," Ph.D. Thesis, Applied Math., California Institute of Technology, 1982.

(173) Roache, P. J. and Steinberg, S., "A New Approach to Grid Generation Using a Variational Formulation," Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 360-370.

(174) Roberts, A., "Automatic Topology Generation and Generalized B-Spline Mapping," Numerical Grid Generation, Ed. by J. F. Thompson, North Holland, 1982, pp. 465-486.

(175) Rubbert, P. E. and Lee, K. D., "Patched Coordinate Systems," Numerical Grid Generation, Ed. by J. F. Thompson, North Holland 1982, pp. 235-252.

(176) Saltzman, J. and Brackbill J., "Applications and Generalizations of Variational Methods for Generating Adaptive Meshes," Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, pp. 865-884.

(177) Saltzman, J., "Variational method for Generating Meshes on Surfaces in Three-Dimensions," Journal of Computational Physics," Vol. 63, 1986, pp. 1-19.

(178) Sanz-Serna, J. M. and Christie, I., "A Simple Adaptive Technique for Nonlinear Wave Problems," Journal of Computational Physics, Vol. 67, No. 2, pp. 348-360.

(179) Saunders, B. V., "Algebraic Grid Generation Using Tensor Product B-Splines," Old Dominion University, Ph.D Thesis, 1985, also NASA CR 177968, 1985.

(180) Sawada, K. and Takanashi, S., "A Numerical Investigation on Wing/Nacelle Interferences of USB Configuration," AIAA-87-0455, 25th Aerospace Sciences Meeting, Reno, 1987.

(181) Schwartz, A. L. and Connett, W. C., "Evaluating Algebraic Adaptive Grid Strategies," Numerical Grid Generation in Computational Fluid Dynamics, Ed. by J. Häuser and C. Taylor, Pineridge Press, 1986, pp. 163-174.

(182) Seibert, W., "An Approach to the Interactive Generation of Block Structured Volume Grids using Computer Graphic Devices," Numerical Grid Generation in Computa- tional Fluid Dynamics, Ed. Haüser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(183) Seidl, A. and Klose, H., "Numerical Conformal Mapping of a Towel-Shaped Region onto a Rectangle," SIAM Journal of Scientific and Statistical Computing Vol. 6, No. 4, 1985, pp. 833-842.

(184) Senoo, Y. and Yoshiyuki, N. , "A Blade Theory of an Impeller with an Arbitrary Surface of Revolution," Transactions ASME Journal of Engineering for Power, 1971, pp. 454-460.

(185) Shaw, J., Forsey, N.P., Weatherill, N. P. and Rose, K. E., "A Block Structured Mesh Generation Technique for Aerodynamic Geometries," Numerical Grid Genera-

tion in Computational Fluid Dynamics, Ed. Hãuser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(186) Smith, R. E. (Ed.), Numerical Grid Generation Techniques, NASA Conference Publication 2166, NASA Langely Research Center, 1980.

(187) Smith, R. E., "Algebraic Grid Generation," Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, pp. 137-170.

(188) Smith, R. E. and Weigel, B. L., "Analytical and Approximate Boundary-Fitted Coordinate Systems for Fluid Flow Simulation," AIAA Paper No. 80-0192, 1980.

(189) Smith, R. E. and Eriksson, L. E., "Algebraic Grid Generation," Computer Methods in Applied Mechanics and Engineering, to appear.

(190) Soni, B. K., "Two and Three Dimensional Grid Generation for Internal Flow Applications in Computational Fluid Dynamics," Proceedings of the AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, 1985, pp. 351-359.

(191) Sorenson, R. L. and Steger, J. L., "Numerical Generation of Two-dimensional Grids by the Use of Poisson Equations with Grid Control at Boundaries," Numerical Grid Generation Techniques, Ed. by R. E. Smith, 1980, pp. 449-461.

(192) Sorenson, R. L., "A Computer Program to Generate Two-Dimensional Grids about Airfoils and Other Shapes by the Use of Poisson's Equation," NASA TM-81198, NASA Ames, 1980.

(193) Sorenson, R. L., "Grid Generation by Elliptic Partial Differential Equations for a Tri-Element Augmentor-Wing Airfoil," Numerical Grid Generation, Ed. by J. F. Thompson, North Holland, 1982, pp. 653-665.

(194) Sorenson, R. L., "Elliptic Generation of Composite Three-dimensional Grids about Realistic Aircraft," Numerical Grid Generation in Computational Fluid Dynamics, Ed. by J. Hãuser and C. Taylor, Pineridge Press, 1986, pp. 353-371.

(195) Sritharan, S., and Smith, P. W., "On the Grid Generation Methods by Harmonic Mappings on Plane and Curved Surfaces," ICASE Report 84-12, 1984.

(196) Starius, G., "Composite Mesh Difference Methods for Elliptic Boundary Value Problems," Numerische Mathematik, Vol. 28, 1977, pp. 243-258.

(197) Starius, G., "Numerical Treatment of Boundary Layers for Perturbed Hyperbolic Equations," Dept. of Computer Science Report No. 69, Uppsala University, Sweden, 1978.

(198) Starius, G., "On Composite Mesh Difference Methods for Hyperbolic Differential Equations,", Numerische Mathematik, Vol. 35, 1980, pp. 241-255.

(199) Starius, G., "Construction of orthogonal Curvilinear Meshes by Solving Initial Value Problems," Numerische Mathematik, Vol. 28, 1977, pp. 25-48.

(200) Steger, J. L. and Benek, J. A., "On the Use of Composite Grid Schemes in Computational Aerodynamics," Computer Methods in Applied Mechanics and Engineering, to appear.

(201) Steger, J. L., "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," AIAA Journal, Vol. 16, No. 7, 1978, pp. 679-686.

(202) Steger, J. L., Sorenson, R., "Automatic Mesh-Point Clustering near a Boundary in Grid Generation with Elliptic Partial Differential Equations," Journal of Computational Physics, Vol. 33, 1979, pp. 405-416.

(203) Steger, J. L., Sorenson, R. L., "Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates," Numerical Grid Generation Techniques Ed. by R. E. Smith, NASA CP 2166, 1980, pp. 463-478.

(204) Steger, J. L. and Chaussee, D. S., "Generation of Body-Fitted Coordinates using Hyperbolic Partial Differential Equations," SIAM Journal of Scientific and Statistical Computing, Vol. 1, 1980, pp. 431-437.

(205) Steger, J. L., Dougherty, F. C. and Benek, J. A., "A Chimera Grid Scheme," Advances in Grid Generation, Ed. by K. N. Ghia and U. Ghia, 1983, pp. 59-70.

(206) Steinberg, S. and Roache, P. J., "Symbolic Manipulation and Computational Fluid Dynamics", Journal of Computational Physics, Vol. 57, 1985, pp. 251-284.

(207) Steinberg, S. and Roache, P. J., "Variational Grid Generation," Numerical Methods in Partial Differential Equations, Vol. 2, 1986, pp. 71-96.

(208) Steinhoff, J., "Blending Method for Grid Generation," Journal of Computational Physics, Vol. 65,No. 2, 1986, pp. 370-385.

(209) Steinhoff, J., "A Recursive Grid Generation Methods for Fighter Configurations," submitted to Applied Mathematics and Computation.

(210) Strang, G. and Fix, G., An Analysis for the Finite-Element Methods, Prentice Hall, Englewook Cliffs, N.J., 1973.

(211) Thacker, W.C., "A Brief Review of Techniques for Generating Irregular Computational Grids," International Journal of Numerical Methods in Engineering, Vol. 15, 1980, pp. 1335-1341.

(212) Theodorsen, T. and Garrick, I. E., "General Potential Theory of Arbitrary Wing Sections," NACA TR 452, 1933.

(213) Thomas, P. D., “Numerical Generation of Composite Three-Dimensional Grids by Quasilinear Elliptic Systems,” Numerical Grid Generation, 1st Ed., New York, North Holland, 1982, pp. 667-686.

(214) Thomas, P. D., “Numerical Generation of Composite Three-Dimensional Grids by Quasilinear Elliptic Systems,” Numerical Grid Generation, 1st Ed., New York, North Holland, 1982, pp. 667-686.

(215) Thompson, J. F., Thames, F. C., Mastin, C. W., “Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies,” Journal of Computational Physics, Vol. 15, 1974, pp. 299-319.

(216) Thompson, J. F., Thames, F. C., Mastin, C. W. and Shanks, S. P., “Use of Numerically Generated Body-Fitted Coordinate Systems for Solution of the Navier-Stokes Equations,” Proceedings of the Second AIAA Computational Fluid Dynamics Conference, 1975, pp. 68-80.

(217) Thompson, J. F., Thames, F. C., Hodge, J. K., Shanks, S. P., Reddy, R. N., and Mastin, C. W., “Solutions of the Navier-Stokes Equations in Various Flow Regimes on Fields Containing any Number of Arbitrary Bodies Using Boundary-Fitted Coordinate Systems,” Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, Vol. 59, Springer-Verlag, 1976, p. 421.

(218) Thompson, J. F., Thames, F. C. and Mastin, C. W., “'TOMCAT' – A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing any Number of Arbitrary Two-Dimensional Bodies,” Journal of Computational Physics, 1977, p. 245.

(219) Thompson, J. F. and Mastin, C. W., “Grid Generation Using Differential Systems Techniques,” Numerical Grid Generation Techniques,” Ed. by R. E. Smith, NASA CP2166, 1980, pp. 37-72.

(220) Thompson, D. S., “A Mesh Embedding Approach for Prediction of Transonic Wing/Body/Store Flow Fields,” NASA CP-2201, 1981, pp. 403-427.

(221) Thompson, J. F., Warsi, Z. U. A. and Mastin, C. W., “Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations – A Review,” Journal of Computational Physics, Vol. 47, 1982, pp. 1-108.

(222) Thompson, J. F., “Elliptic Grid Generation,” Numerical Grid Generation, Ed. by J. F. Thompson, North-Holland, 1982, pp. 79-105.

(223) Thompson, J. F. (Ed.), Numerical Grid Generation, North-Holland 1982. (Also published as Vol. 10 and 11 of Applied Mathematics and Computation, 1982).

(224) Thompson, J. F., "Grid Generation Techniques in Computational Dynamics," AIAA Journal, Vol. 22, 1984, pp. 1505-1523.

(225) Thompson, J. F. and Warsi, Z. U. A. and Mastin, C. W., Numerical Grid Generation, Foundations and Applications, North-Holland, 1985.

(226) Thompson, J. F., "A Survey of Dynamically Adaptive Grids in the Numerical Solution of Partial Differential Equations," Applied Numerical Mathematics, Vol. 1, 1985, pp. 3-27.

(227) Thompson, J. F., "Composite Grid Generation for General 3D Regions," Numerical Grid Generation in Computational Fluid Dynamics, Ed. Hāuser and Taylor, Proceedings of the International Conference in Landshut, West Germany, 1986.

(228) Thompson, J. F., "Composite Grid Generation for General 3D Regions," Computer Methods in Applied Mechanics and Engineering, to appear.

(229) Trease, H. E., "A Two-Dimensional Free-Lagrangian Hydrodynamics Model", Ph.D. Thesis, University of Illinois, Urbana-Champaign, 1981.

(230) Trease, H. E., "Three-Dimensional Algorithms for Grid Restructuring in Free-Lagrangian Hydrodynamics," The Free-Lagrange Method, Lecture Notes in Physics, Springer-Verlag, New York, 1985 pp. 145-157. Lecture Notes in Physics, Vol. 238, Springer-Verlag, 1985, pp. 87-113.

(231) Trefethen, L. N. (Ed.) Numerical Conformal Mapping, North Holland, Amsterdam, 1986.

(232) Venkatapathy, E. and Lombard, C. K., "Flow Structure Capturing on Overset Patched Meshes," AIAA-85-1690, 1985.

(233) Visbal, M. and Knight, D., "Generation of Orthogonal and nearly Orthogonal Coordinates with Grid Control near Boundaries," AIAA Journal, Vol. 20, No. 3, 1982, pp. 305-306.

(234) Warsi, Z. U. A., Thompson, J. F., "Machine Solutions of Partial Differential Equations in the Numerically Generated Coordinate Systems," Report MSSU-EIRS-ASE-77-1, Missisipi State University, Missisipi State, 1976.

(235) Warsi, Z. U. A., "Tensors and Differential Geometry Applied to Analytic and Numerical Coordinate Generation, MSSU-EIRS-81-1, Aerospace Enginnering, Missisipi State University, 1981.

(236) Warsi, Z. U. A., "Basic Differential Models for Coordinate Generation," Numerical Grid Generation, 1st Ed., New York, North Holland, 1982, pp. 667-686.

(237) Warsi, Z. U. A. and Thompson, J. F., "A Noniterative method for the Generation of Orthogonal Coordinates in Doubly-Connected Regions," Numerical Grid Generation Techniques, Ed. by R. E. Smith, 1980, pp. 519-544, also in Mathematics of Computation, Vol. 38, 1982, pp. 501-516.

(238) Warsi, Z. U. A., "Numerical Grid Generation in Arbitrary Surfaces Through a Second Order Differential-Geometric Model," Journal of Computational Physics," Vol. 64, 1986, pp. 82-96.

(239) Wedan, B. and South, J. C., "A Method for Solving the Transonic Full-Potential Equation for General Configurations," Proceedings of the 6th AIAA Computational Fluid Dynamics Conference, Danvers, MA, 1983, pp. 515-526.

(240) White, A. B.. Jr., "On Selection of Equidistributing Meshes for Two-Point Boundary-Value Problems," SIAM Journal of Numerical Analysis, Vol. 16, 1982, pp. 472-502.

(241) White, A. B., Jr., "On the Numerical Solution of Initial Boundary-Value Problems in One Space Dimension," SIAM Journal of Numerical Analysis, Vol. 19, 1982, pp. 683-697.

(242) Winkler, K.-H. A., "A Numerical Procedure for the Calculation of Nonsteady Spherical Shock Fronts with Radiation," Ph.D Thesis, University of Gōttingen. Issued as Technical Report MPI-PAE-Astro 90 by the Max-Planck-Institut für Physik und Astrophysik, Munich; English Translation: Lawrence Livermore Laboratory, Report UCRL-Trans-11206. University of California, Livermore, 1976.

(243) Winkler, K-H. A., Norman, M. L., and Mihalas, D., "Implicit Adaptive Grid Radiation Hydrodynamics," Multiple Time Scales, Ed. by J. U. Brackbill and B. I. Cohen, Academic Press, 1985, pp. 141-184.

(244) Winkler, K-H. A., Mihalas, D., and Norman, M. L., "Adaptive Grid Methods with Asymmetric Time-Filtering," Computer Physics Communications, Vol. 36, 1985, pp. 121-140.

(245) Winslow, A., "Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh," Journal of Computational Physics, Vol. 2, 1967, pp. 149-172.

(246) Yanenko, N. N., Kovenya, V. M., Lisejkin, V. D., Fomin, V. M., Vorozhtsov, E. V., "On some Methods for the Numerical Simulation of Flows with Complex Structure," In Lecture Notes in Physics, Vol. 90, Springer-Verlag, 1978, pp. 565-578.

(247) Yeung, R. W., "Numerical Methods in Free Surface Flows," Annual Review of Fluid Mechanics, Vol. 14, 1982, pp. 395-442.